ScanVue Lite
User Guide

(U.S. Patent No. 6,213,394 B1)

©2000,2001,2002,2003,2004,2005 INDUSTRIAL ELECTRONIC ENGINEERS, INC.
7740 Lemona Avenue

Industrial Electronic Engineers, Inc.

Van Nuys, CA 91409

NOTICES

This document contains proprietary information shall not be used or reproduced or its contents disclosed in whole or in part, without the prior written consent of Industrial Electronic Engineers, Inc.

Industrial Electronic Engineers, Inc. reserves the right to make improvements to the software described in this documentation at any time and without notice. The information contained herein is subject to change without notice and should not be construed as a commitment by Industrial Electronic Engineers, Inc.

The software described in this document is provided as a licensed item in conjunction with Industrial Electronic Engineers, Inc. equipment. It may not be copied or distributed for use on other than the machines it was licensed for.

DISCLAIMER OF ALL LIABILITIES

Industrial Electronic Engineers, Inc. shall have no liability or responsibility to you or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by this documentation or the hardware or software described in it. This includes but is not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such software or computer programs.

All information in this manual was deemed correct at the time of printing. Updated versions of this manual will be published periodically and may be downloaded in Abode Acrobat (PDF) format from IEE's website WWW.IEEINC.COM.

Copyright © 2004 Industrial Electronic Engineers, Inc.
All Rights Reserved.

NO PART OF THIS PUBLICATION MAY BE REPRODUCED
WITHOUT THE WRITTEN PERMISSION OF
INDUSTRIAL ELECTRONIC ENGINEERS, INC.

STANDARDS CERTIFICATION

The SVLite product described in this manual has been fully tested and certified by an independent testing laboratory and is compliant with the following international standards.

- UL Standard 60950 (ITE) listed product.
- CSA standard C22.2 No. 950 recognized product.
- CFR Title 47 part 15, Class B
- EN55022:1998 Class A, EN55024:1998 (73/23/EEC)
- EN61000-1:2001, EN61000-3:2001 (89/336/EEC)

The Intermec line scanner mounted in the underside of the SVLite Price Verifier unit complies with the following standards and regulatory requirements.

- CUL recognized component
- VDE

TABLE OF CONTENTS

TABLE OF FIGURES vi

Chapter 1—Introduction 1

What's in this User Guide 1
ScanVue Lite Models 1
Description 1
Technical Knowledge Required 2
Installation 2
Application Program Interface (API) 2
Technical Assistance 3
Chapter 2—Installing SVLite 4

Items Shipped in Box 4
General Wiring Considerations 4
Mounting the SVLite Unit 4
Attaching a Serial Printer 6
Configuring the Serial Port 7
Sending Data to the Printer 7
Co-Located Printer 9
Chapter 3—Installing the Software 10

Server Install 10
Unit Configuration 11
Chapter 4—Network Configuration 12

Introduction 12
Network Activity 12
SVLite Configuration 13
Quick Start 13
Configuration Information Screens 13
UnitConfig Program 15
Using Unit Configuration 15
Configuration Rules 17
Named Servers 17
Network Diagnostics 18
Diagnostic Configurations 18
Diagnostic Screens and Messages 19
Chapter 5—Interfacing to the Back Office Server 21

Software Developers Kit (SDK) 22
ProductInfo Protocol Description 23
Protocol Implementation Rules 24
SVLite Implementation Rules 25
SVLite Supported Modes 26
Fixed Unit Identification 26
User Configurable Unit identification 26
Setup for Windows Networking (SMB) 26
Setup for FTP, QFX, NTP, PRODUCTINFO 26

Product Query configuration 27
Presentation configuration 27
Real Time Clock Display (LCD model only) 28
Miscellaneous configuration 28
Command modes 30
Programmatic Modes 30
QFX Quick File Transfer Protocol 32
Using Graphics Characters (LCD Display) 33
Using Input Devices 34
Push Buttons 34
Start Event and End Event Masks 34
Structure of EVENT packet 35
Examples (shown in hex form) 35
SVLite420 Vacuum Fluorescent Display 36
Description 36
Operation 36
Text Slideshow 37
Changing Language Character Sets 37
Customer Messages 38
Changing Customer Messages 38
Appendix A—Configuring with Barcodes 40

Barcode Reader Settings 40
Recommended Scanner Settings 40
Configuring SVLite® 41
Appendix B—SVLite Initialization File 45

Overview 45
Real Time Clock 45
Rules for the scanvue.ini File 46
Sections & Commands 46
Example of Initialization File 48
Appendix C—Creating a Slideshow 50

Creating a Slideshow File 50
Appendix D—Font and Language Sets 51

Graphics (LCD) Display Models 51
Vacuum Fluorescent (VF) Display Models 52
Appendix E—ProductInfo Protocol 56

Appendix F—Mounting Bracket Outline 63

Appendix G—Application Notes 67

Creating Customer Messages on VF Display Models 67

Network Messages 67
Greeting Message 69
Appendix H—Expanded Text API 70

API Notation 70
Defining a Text Box 70
Writing to a Text Box 71
Text Timeout 71

Appendix J—Local Lookup Database 72

TABLE OF FIGURES

Fig 1—Wall mount bracket 5
Fig 2—Back of SVLite Price Verifier 6
Fig 3—Wiring a Serial Printer 8
Table 1—Serial Port Y Cable Part Numbers 8
Fig 4a,b,c,d—Configuration Status Screens, VF model 14
Fig 5a,b—Configuration Status Screens, LCD model 15
Fig 6—Unit Configuration (UnitConfig) Screen 17
Fig 7—Diagnostic Configurations 18
Fig 8—Simplified SVLite System Diagram 21
Table 2 Tokens and Data Types 24
Table 3 Event Start (pushbutton) Masks 28
Table 4 VFD Display Language Selection 30
Table 5 Pushbutton Masks 35
Fig 9—Greeting Screen 36
Fig 10—Item Description & Price Display 37
Table 6 VFD Display Selecting Language 37
Table 7 —Barcode Configuration Labels 44
Fig 11—Real Time Clock Display (LCD only) 45
Table 8—Default ASCII Character Set For LCD Display 51
Table 9—Default ASCII Character Set for VF Display 53
Table 10—European Character Set for VF Display 54
Table 11—Katakana Character Set for VF Display 54
Table 12—Cyrillic Character Set for VF Display 55
Table 13—Hebrew Character Set for VF Display 55
Fig 12—Wall Mount Bracket 63
Fig 13—Co-Located Printer Outline Bracket 66
Fig 14—UnitConfig Screen, msgChecking Mode 67
Fig 15—Default English Greeting Message 69
Fig 16—Spanish Greeting Message 69
Change history 73

Chapter 1—Introduction

What's in this User Guide

This user guide provides instructions for installation and operation of the IEE ScanVue Lite (SVLite) Price Verifier. A description of the basic functions and features of the hardware and software as well as how to physically install the unit, set it up to operate on your network and interface SVLite to a back office server application using its Application Programming Interface (API).

SVlite is a scaled down version of its bigger brother, ScanVue5. The VF model will run the identical server application as its bigger brother and the monochrome LCD model will run the same code as the ScanVue5 color AMLCD version but doesn't support all the features.

SVLite is able to receive updated software revisions downloaded across the network. The current software version number is displayed on the first status screen appearing after powering up the unit. The status screen can be displayed directly by scanning a special barcode (see Appendix A); the host can also query the unit for the same information.

Software updates and later versions of this manual will be made available on IEE's web server at www.ieeinc.com as they are released. Revision A of the User Guide is current with software versions 1.7 and later.

ScanVue Lite Models

The low cost SVLite comes in 2 models—a 4x20 Vacuum Fluorescent (VF) text only display or a 320x240 pixel (¼VGA) blue/white monochrome LCD graphics display. Both versions incorporate a linear CCD imaging barcode scanner.

The 4x20 VF model has 4 languages built-in (English/European, Cyrillic, Hebrew and Katakana) whereas the LCD model can make use of hundreds of different downloadable fonts and can display graphics images. SVLite derives its power from a fully compliant IEEE 802.3af power-over-Ethernet (PoE) link.

Both models have a single serial RS232 port for a printer or other serial device. Two options are available—4 programmable pushbuttons and a DC power wall mount or desk mount power supply for users that do not have PoE.

A model with a co-located Citizen printer is available. This model was specially developed for users needing the price/description printed on a label for states with regulations that require this.

Description

SVLite is a low cost price verifier designed for retail store customers to check the price of any UPC bar coded product without having to leave the aisle or shelf area. With its optional label printer, SVLite meets the requirements of US and

Canadian state governments that have legislated the use of price verifiers in certain retail stores.

The contemporary housing design merges well with almost any store décor and custom color combinations are available if the units are ordered in sufficient quantities. SVLite attaches to a mounting bracket that can be located on a wall, in an aisle, on a column or endcap or at a POS location.

The electronics package is housed in a high impact ABS injection molded case. The unit meets most worldwide regulatory safety and EMC standards including UL1950, CSA22.2, FCC and CE.

Technical Knowledge Required

Installation

The installation of SVLite is a simple process for a person familiar with installing and maintaining local area networks (LAN's). Although several software tools have been provided to help with the network setup, this manual is not intended to be a training guide for novice network installers. It is assumed the installation will be done by a person having a sufficient level of technical expertise with LAN hardware and software to understand the content of this manual and complete the job with minimal outside help. A system or network administrator has the knowledge to perform the installation with ease. IEE can provide additional technical assistance in getting the system running, if required.

Application Program Interface (API)

An API that provides the interface between SVLite and the host computer application is described in detail in Chapter 4. Most API programs and utilities are written in ANSI "C" and can be integrated with the customers server based code using the appropriate ANSI 'C' compiler. Some OCX's are provided for users wanted to integrate with VB. The database application and its interface to SVLite® will generally reside on a local networked host computer (sometimes called the network server). Any hardware or software platform may be used as long as the network transport protocol is TCP/IP. Platforms currently running ScanVue API include Wintel hardware running Windows NT/2000/ME/XP, VAX running VMS, AS400, Unix, Oracle, SQL…

The database application program is responsible for receiving a request (or query) from the scanner, retrieving the price and description from the database, and returning that data to the unit that initiated the request. Development and maintenance of any host computer based application program required to access a product–information database is the sole responsibility of the end user or their system integrator. IEE provides an SDK including some license free "C" source code to assist the end user in writing applications on their host machine.

Software versions V1.7 and later can also retrieve price/description data from a local database file. This feature can provide very rapid response to a scan and does not rely on the network to deliver the information. The end user is responsible for updating the local database file periodically. The local database

has a physical size restriction of approximately 1200 items. This feature and its operation is described in detail in Appendix J.

Technical Assistance

If you require technical assistance in bringing up your SVLitePrice Verifier, please call (818) 787–0311 and ask for Technical Support. You may also e–mail questions to TechSupport@IEEInc.Com. Please be prepared to discuss your configuration and setup parameters.

Chapter 2—Installing SVLite

Items Shipped in Box

- SVLite price verifier with wall mount bracket.
- SDK CD-ROM (software developers kit). Includes programs for setting up the SVLiteunit and the server, API source code, demo programs, font files, user guides, and useful third-party utilities.
- Optional accessories:
  - Universal Input DC Power Supply (IEE P/N 90209-21)
  - Cable for Epson P90L or Citizen CBM1000II Label Printer (IEE P/N 38484-XX)
  - Cable for Zebra LP2824 Label Printer or NCR K590 printer (IEE P/N 38254-XX)
  - Cable for NCR 7137 Thermal Printer (IEE P/N 38094-XX).
  - Cable for Symbol Cobra or NCR 7837 hand scanner (IEE P/N 37915-XX).

General Wiring Considerations

While SVLite is designed to withstand power and data line surges, spikes and other anomalies in accordance with IEC and CE specifications it may not survive a direct lightning strike. In parts of the USA and worldwide where there is a high likelihood of thunderstorm activity it is good practice to install lightning surge protectors on all power and data lines. SVLite should be treated like any other network and computer product installed in your facility.

Mounting the SVLite Unit

The SVLite price verifier slides onto a steel mounting bracket attached to a wall or column by 6 screws, then secured in place by tightening 2 hidden screws on the bottom of the unit.

A dimensioned outline drawing of the mounting bracket is shown in Appendix G.

1. Position the mounting bracket on the wall or column with the rectangular opening up and mark the locations of the 6 countersunk mounting holes and the one large hole for wiring access.
2. Drill the 6 small mounting holes for an appropriate size and type of fastener depending on the type of wall being used to mount the PV.
3. If the CAT5 network cable is to be run inside the wall, drill out the large hole to allow the network cable connector and the DC power supply connector (if you are not using PoE) to be pulled through. Provided you have a hollow wall, this method provides the neatest and most secure installation.
4. Plug the network cable connector into the RJ45 socket and the round 8 pin DIN power supply connector into its socket in the back of the unit (if the external power supply is being used).

5. If the cables run inside the wall, push all the excess cable back inside the wall before sliding the PV onto its bracket. If the cables run on the surface of the wall, route them through the 2 smaller U shaped cable slots in the bottom rear of the housing before sliding the PV onto its bracket. **Do not run the cables in the larger U slots or you will not be able to secure the unit to the bracket properly.**

6. Slide the PV over the mounting bracket aligning the 4 protruding tabs with the 4 cutouts on the PV rear chassis bracket then push the PV towards the wall and slide it down until the tabs fully engage the slots in the PV rear chassis bracket. When properly engaged, the PV will lie flush with the wall and the top of the PV will be aligned with the top of the mounting bracket.
7. Tighten up the two Phillips head screws visible through the 2 larger U slots in the bottom of the rear housing—this will lock the PV to the mounting bracket and prevent it from being removed from the wall.

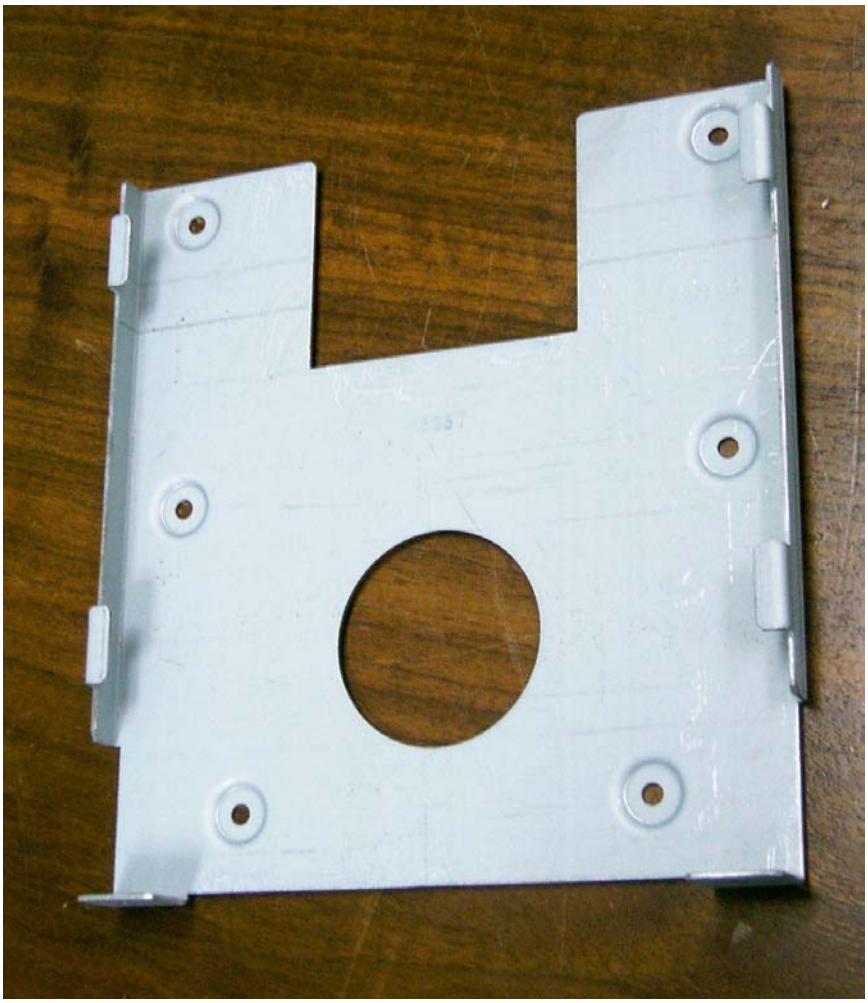


Fig 1—Wall mount bracket

Fig 2—Back of SVLite Price Verifier

Attaching a Serial Printer

A label printer can be attached to the SVLite price verifier serial port by plugging an optional Y cable into the DIN8 connector in the back of the unit. The serial port (serial port 2) shares the DIN8 connector with the DC power input and the Y cable splits the power and serial port connections to separate connectors. If PoE is utilized, you can hide the power connector leg of the Y cable inside the back of the housing.

The factory default setting for serial port 2 is **OUTPUT** which will work for most printers. If the printer requires data to flow back to the host—for instance, it has a paper-low sensor that is monitored by the host—change the setting to

BIDIRECTIONAL using UnitConfig or the modeset commands. The port may also be disabled by setting it to **NONE**.

In **OUTPUT** or **BIDIRECTIONAL** mode, serial port 2 is a transparent RS232 data port. The price verifier receives pre-formatted data and control codes from a printer driver resident on the network host, which it passes, unmodified, through serial port 2 to the printer. Printer drivers are specific to the printer used and must be provided by the user.

Configuring the Serial Port

In **OUTPUT** mode the port parameters are pre-configured to 9600 baud, no parity, 8 data bits, 1 stop bit. In **BIDIRECTIONAL** mode the port parameters are pre-configured to 19200 baud, N, 8, 1. Different manufacturers printers may require a different Y cable or port settings. The serial port parameters must match the printer's serial parameters to be able to communicate properly. A utility program **PortSet** is provided on the CD-ROM to allow new configuration parameters to be pushed to the PV. Check with your sales person before connecting any printer not specifically mentioned in this document.

Use the UnitConfig program or modeset utility to change the serial port configuration to **OUTPUT** or **BIDIRECTIONAL**. If you are using an MS Windows system, open a DOS box and change to the C:\POS directory where modeset is stored and send the 2 following lines. The modeset utility, found on the CD-ROM, must be copied onto your server before these commands can be sent.

C:\POS modeset -iPort2function=OUTPUT {BIDIRECTIONAL} <IP> ↵
C:\POS modeset -c -r <IP> ↵

where <IP> is the unit's IP address.

Sending Data to the Printer

The host server must use nominal-mode ProductInfo packets to send data to the printer (see Chapter 5). The packets required are **Data** type, **Special Text** sub-type.

An example of a typical message is:

Byte Count	Value (Oh)	Meaning
0-3	00 00 00 13	Length of packet (19 bytes)
4-7	44 41 54 41	'DATA' Data type
8-11	53 50 45 43	'SPEC' Special Text sub-type
12-18	48 65 6C 6C 6F 21 0A	'Hello!<LF>' Text sent to printer

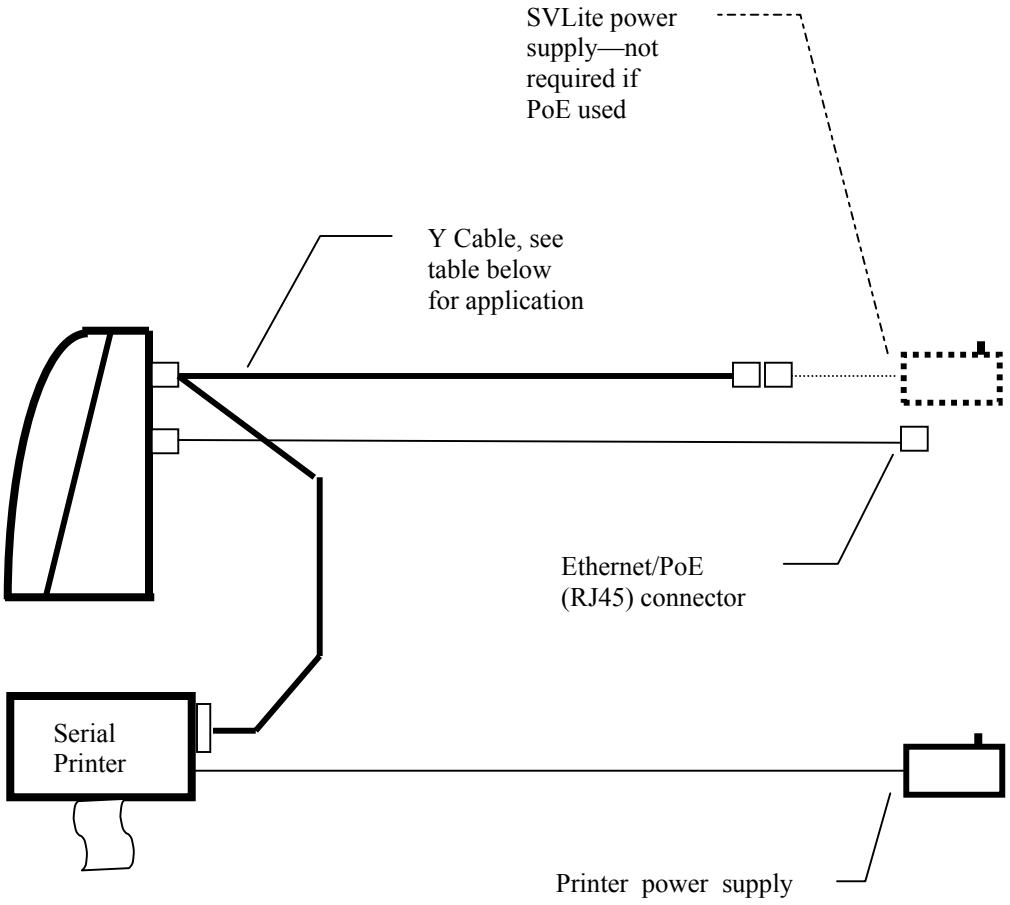

SVLite power supply—not required if PoE used
Y Cable, see table below for application
Ethernet/PoE (RJ45) connector
Serial Printer
Printer power supply

Fig 3—Wiring a Serial Printer

Printer or Hand Scanner	Cable Part No.
NCR 7137 Thermal Printer	38094-xx
NCR K590 Printer	38254-xx
Zebra LP2824 Label Printer	38254-xx
Epson P90L Thermal Label Printer	38484-xx
Citizen CBM 1000II Thermal Label Printer	38484-xx
Symbol Cobra Hand Scanner	37915-xx
NCR 7837 Hand Scanner	37915-xx

Table 1—Serial Port Y Cable Part Numbers

Co-Located Printer

This model is provided with a Citizen CBM1000II roll/label printer mounted side by side with Scanvue on a common wall mount bracket. Scanvue and the printer are powered by a single power supply. In this hardware configuration, the amount of current drawn by the printer+Scanvue precludes using PoE and the serial port is already configured and dedicated to the printer.

If the autocutter is used to separate labels, the labels must be spaced at least 13mm apart for the sensor to located the correct cutting position.

Chapter 3—Installing the Software

We highly recommend you install the latest updates or service packs to the operating system you are using before installing SVLite programs onto your server or host computer. The UnitConfig program especially requires a minimum level of Internet Explorer to be running. On Windows 98, IE Explorer 6.0 or later and NT or XP require IE Explorer 5.50 or later.

The ***SetupServer*** program on the CD–ROM must be installed on your Windows network server to create a specific SVLite® directory called **POS** on your servers' hard disk. Share this directory and copy the default **scanvue.ini** file into **POS**. SVLite® units look for this file in the **POS** share when they boot up and will not operate if they can't find it.

You should also install ***UnitConfig***, the easiest to use, graphical table oriented program for setting configuration parameters. The command line version of this program is called ***Modeset*** and runs in a DOS box. *S*canning barcodes is yet a 3rd way to set the configuration and sometimes may be the only way if the PV is not communicating with the network.

You can also install ***UnitConfig*** on a desktop or laptop computer for setting up SVLite units locally without a large network complicating it. You will need a network interface card (NIC) in your PC configured to communicate with SVLite and an optional SVLite power supply brick as your PC will not be PoE.

For servers that are running Operating Systems other than Windows, sources are provided for the server software. There are also freely available drivers to allow any operating system to act as a Windows–type server.

Important: the shared ScanVue directory \POS and the scanvue.ini file must be on the server and the SVLite price verifiers must be able to find them when they power up. If you want to change the operating modes (or configuration) of a specific unit anytime after the system is up and running, the **UnitConfig** program is available in the \POS directory.

Server Install

When you place the IEE SDK CD–ROM (p/n 37720-01) into a drive, the autoplay feature will open a window with the folders and program icons. If it doesn't start on its own, click Windows **START** button, then **Run** and enter the CD-ROM drive name (usually d:) in the text box and click **OK**. The CD will open a new window–double click the IEE folder. Click on the IEE icon ***SetupServer*** and the install program will lead you step by step through the process of creating the **POS** directory, sharing it, and copying the scanvue.ini file into the directory.

Server install also creates a font directory under **POS** and loads all the fonts on the CD–ROM into this directory. SVLite has one default font programmed into the unit. This is a 16x32 dot glyph containing all 256 characters of the extended

ASCII character set—thus any Latin based language can be supported from this font set. **Note**: the soft fonts in the font directory are only supported on LCD display models. VF display models have 4 built in hard fonts (see Appendix D) but cannot use the soft fonts.

Unit Configuration

Click on the IEE icon ***UnitConfig*** in the root directory SDK CD–ROM window and the Unit Configuration program will be installed in a subdirectory under POS. This program, which consists entirely of the one screen shown in the section on using **Unit Configuration**, implements the specific set of commands and controls defined in the API for SVLite® (see Chapter 5). The program is written in Visual Basic® and the sources and an OCX are provided on the CD–ROM so users can write directly to this program and make or initiate mode changes from their application program

Chapter 4—Network Configuration

Introduction

SVLite is a networked thin client device that will connect to any hardware or software platform that uses TCP/IP socket protocols for communication. Connecting SVLite to your network requires the same kind of preparation as connecting a workstation to your network. This guide assumes you know how to obtain the relevant information about your existing network configuration and choose the appropriate configuration parameters so your network will recognize SVLite price verifiers. We recommend connecting to your network through switches rather than hubs as the bandwidth and response time is much better.

SVLite has two primary modes. It initially displays a 'Welcome' or 'Scan Here' screen or a simple graphics slide show if the LCD model is being used. After an item is scanned the items price and description is displayed. In order to retrieve this product information, SVLite must connect to a host computer that has a database of all the items in the store that can be looked up. If you have an existing barcode based customer checkout POS system, the database and lookup capability already exists—thus integrating SVLite into this system is a matter of adding the SVLite interface to the POS application.

The four optional buttons on the front panel are individually programmable and uncommitted. Actual examples of programmed button functions include 1) a button to change the primary language displayed 2) a button to print a label in compliance with the laws in some states and 3) a button to print a coupon.

If you are using an MS Windows network, we highly recommend Windows NT, 2000 or XP. Windows 95 and 98 can be used as a server for SVLite but they are no longer supported. NT4 Server and Windows 2000 will provide the best performance in terms of response speed.

Network Activity

SVLite has servers for FTP (port 21), ProductInfo (port 1283), and clients for FTP, SMB (Windows networking), ProductInfo (Product Information Protocol) and QFX (Quick File eXchange). SMB is the default mode.

SVLite ships from the factory with the following default network settings.

IP Address:	DHCP	Sub–Net Mask:	255.255.0.0
Unit ID:	SVLite		
User Name:	GUEST	Password:	(none)
Windows Server:	SVSERVE	Windows Share:	POS

SVLite requires a server (host or back office computer) for the price lookup database. The file server must have the SVLite initialization file (scanvue.ini) in

its **/POS** shared directory (if SMB–based) or the default directory for FTP or QFX, and may also have font and graphic files if the LCD model is used. The file server can be a Windows system, or an FTP or QFX server on any type of hardware or OS provided it runs TCP/IP.

SVLite sends the UPC number read from a barcoded item placed under the scanner to the server on the host computer which uses this number as a key to find the item in the price lookup database. After the item records are retrieved, the host application formats the response and sends it back to the price verifier where it is displayed. The response must be text only if a VF display is being used or can be a combination of text and graphics for an LCD display.

SVLite Configuration

There are three ways to change the configuration of a SVLite unit. Sending commands across the network using the graphical interface UnitConfig program or the command line prompt Modeset program or by using the barcode reader to scan configuration barcodes. Using barcodes does not require the PV be connected to the network. Configuration barcodes that can be printed out are provided in the documentation folder on the CD-ROM. SVLite must be connected to the server before it can be changed with UnitConfig or Modeset.

The easiest way to connect SVLite to a server is to follow the procedure in the 'Quick–Start Guide' provided in the documentation directory on the CD-ROM.

Quick Start

The CD–ROM a document called Universal QuickStart Guide. This document describes how to configure your units to match the network settings using a desktop or laptop PC and a crossover cable or a passive hub (see the last 2 diagrams on page 24). This is the easiest way to bring up the first few units without the added complications of a large network.

Configuration Information Screens

When SVLite boots up, configuration status screens showing the current settings of the unit are displayed sequentially. Figs a,b and c show the status screens for the VF display and Figs d and e for the LCD. Each screen is displayed for 10 seconds and then the unit will display a logo and 'Welcome Scan Your Item Here" message. The status screens can be displayed at any time by scanning the 'info screen 1' and 'info screen 2' bar codes.

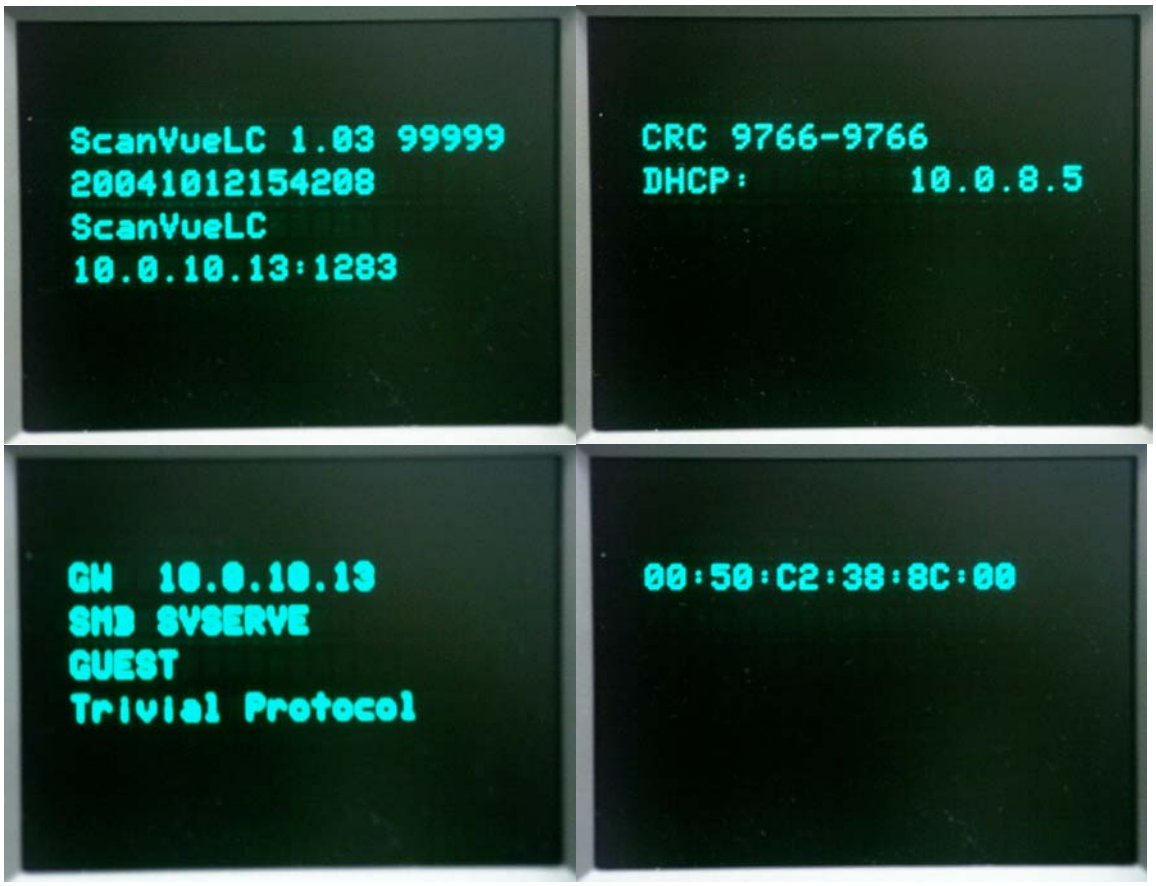


Fig 4a,b,c,d—Configuration Status Screens, VF model

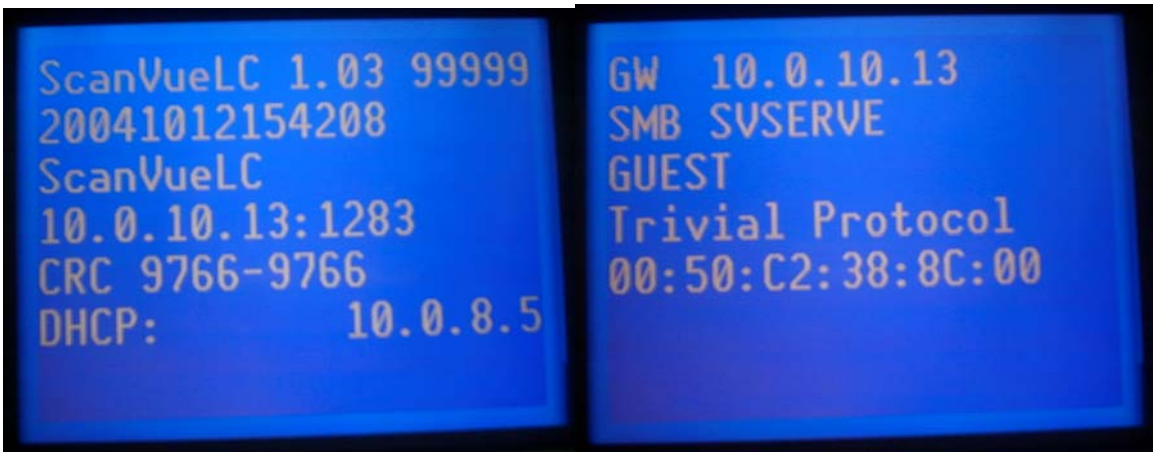


Fig 5a,b—Configuration Status Screens, LCD model

UnitConfig Program

UnitConfig is a VisualBasic® program that runs on the host computer and communicates with SVLite over the network. It provides a simple graphical way to query and configure any SVLite unit and uses the mode commands (Chapter 5) to change settings. This program is available in both source and executable form on the CD-ROM. The UnitConfig program can be installed from the CD–ROM to your host server or offline computer by following the "Quick–Start Guide" instructions. ***UnitConfig*** is the easiest way to change configuration settings.

Modeset is a DOS command line program that does exactly the same thing that UnitConfig does except from a DOS prompt. Modes can be queried or set from a text file or directly from the directory prompt. A DOS box must be open to use modeset. All available modes are fully specified in Chapter 5. Modeset may be more convenient to use within a server application if configuration changes are required as part of the application although an OCX is provided for UnitConfig.

Ex: modeset [mode value]	sets the mode to the specified value
modeset –g [mode]	returns and displays the value of the mode
modeset	lists all non-hidden modes and their values

Using Unit Configuration

When UnitConfig is started, all cells in the table are blank. Enter the IP address of the SVLite unit you wish to change in the ***Unit IP Address*** box. Click the ***Read Modes*** button–all the modes and their contents (values) will be read back from the subject unit and displayed as shown in Fig. 9.

To change a mode— click on and highlight the ***New Content*** box in the row for the mode you want to change. Enter the new value in the box and click the ***Set Mode*** button. If the value is accepted, the box and the button will turn green. If it is not accepted, the box and button will turn red. Make any other changes. Now click the ***Commit*** button, this will commit the change to flash memory in the SVLite unit. If any of the changes are not accepted, the ***Set Mode*** button will turn red and those changes will not be made. Click the ***Restart Unit*** button to

reboot SVLite and save all the changes just made. If you don't click ***Commit*** the changes will only be temporary, which is useful for troubleshooting, and when you reboot the original values will be restored.

Clicking the ***Close*** button will blank the screen including the IP address box and allow a new IP address, hence a new unit to be selected.

Clicking the ***TextReset*** button will apply changes made to text modes so they can be seen immediately. As with other modes, the changes are not stored permanently until the ***Commit*** button is clicked.

The ***SaveToFile*** button will save the setup information to a text file where it can be stored and printed if necessary. This is usually done for troubleshooting or to maintain hard copy records of each unit's configuration.

Clicking on the ***Help*** button brings up a help screen that is a brief overview of how to use the program. Additionally, the light color area at the bottom of the screen will display context sensitive help for each mode as the mode is highlighted. For some modes, the values available are indicated in this area. Other modes (for example ***Serial Number***) cannot be changed and will always turn the ***New Content*** box and ***Set Mode*** button red.

There are many more lines (mode settings) than can be displayed at one time on the screen. Scroll down to get the rest of the modes.

The ***Exit*** button closes the Unit Configuration program.

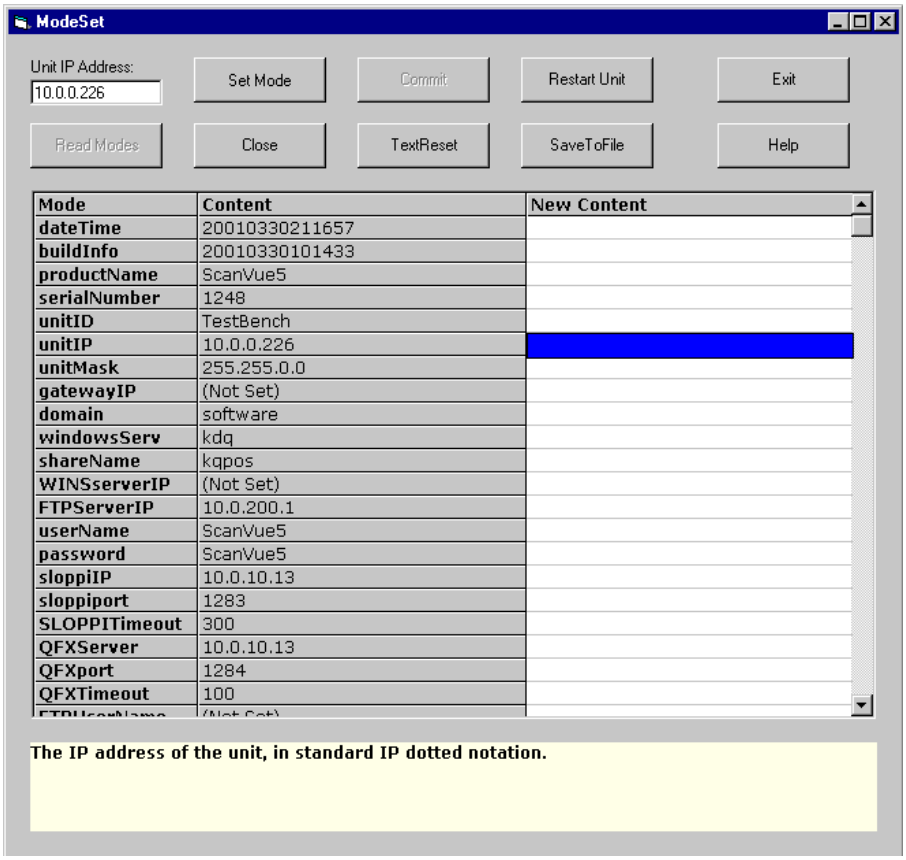


Fig 6—Unit Configuration (UnitConfig) Screen

Configuration Rules

1. Set the ***UnitIP*** to **DHCP** to have SVLite obtain its IP Address from a network DHCP server. This is the factory default setting. If you are not using a DHCP server, a fixed IP Address must be entered.
2. You may specify any or all of the types of file servers. The one that will be used is determined by the ***ServerType*** mode.
3. The value **–default–** has a special meaning: it will return that mode to the factory default value.

Named Servers

Named servers allows the modes **SloppiIP** and **QFXserver** to be entered as a server name or as the server IP address.

Network Diagnostics

Diagnostic Configurations

If you are having problems connecting SVLite to your network with, we highly recommend that you download **EtherHelp®** and its technical documentation from the www.wildpackets.com web site. **EtherHelp®** is the resident piece of a remote network diagnostic tool (network sniffer) IEE owns and uses for troubleshooting network problems. Three useful network configurations for troubleshooting are shown below. There is no charge for the use of this tool.

Run Etherhelp and submit the results to IEE Technical Support for analysis. Note: it is very rare to have to resort to this tool for help—usually network issues can be resolved over the phone quite easily.

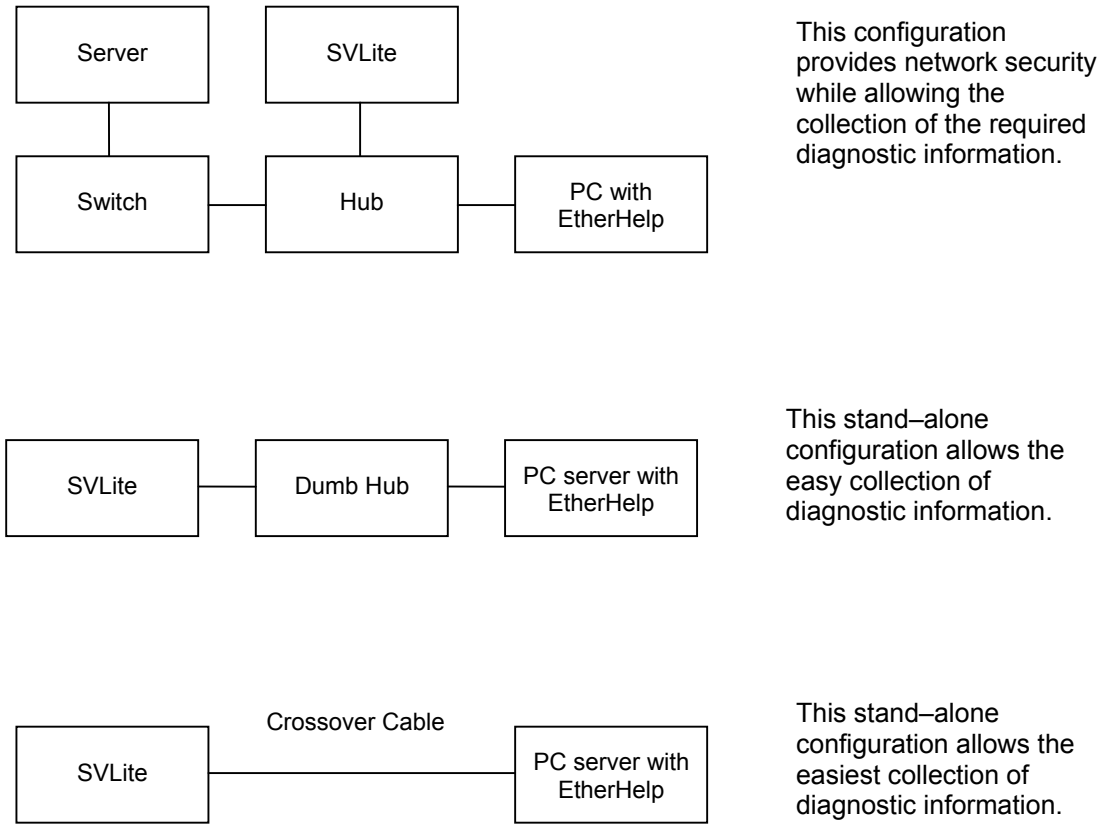


Fig 7—Diagnostic Configurations

The hub must be a “dumb” or passive hub i.e., one that does not do any routing. If a switch or intelligent hub is used, EtherHelp® will not be able to capture all of the packets from the ScanVue. When in doubt, the crossover cable is the best solution.

Diagnostic Screens and Messages

SVLite software has a built in diagnostic that will automatically attempt to determine and resolve network connectivity problems when the unit is powered up. By their very nature diagnostics are not completely fool proof, but they are powerful tools and many times can pin point problems or provide a strong clue to the actual problem. The host must be able to ping the unit for the diagnostics to be meaningful.

The first line of the display will indicate the type of connection being attempted– Windows networking (SMB), FTP, or Quick File Exchange (QFX). The unit will display, at most, one diagnostic screen between restarts. If you have multiple problems they will show up one at a time.

The second line indicates the general class of error, **Internal**, **Connect**, or **Transfer**. The third line will name the specific error, and if a file name is involved, it will be displayed on the fourth line.

Internal Error

An **Internal** class error usually relates to an SVLite device failure and should never be seen. If the unit can be pinged it is working well enough to eliminate device failure as a cause of not connecting.

Transfer Error

Transfer class errors indicate problems with transferring data over the network after a connection has been established to the host. Again, the explanations provided are not necessarily the only possibilities.

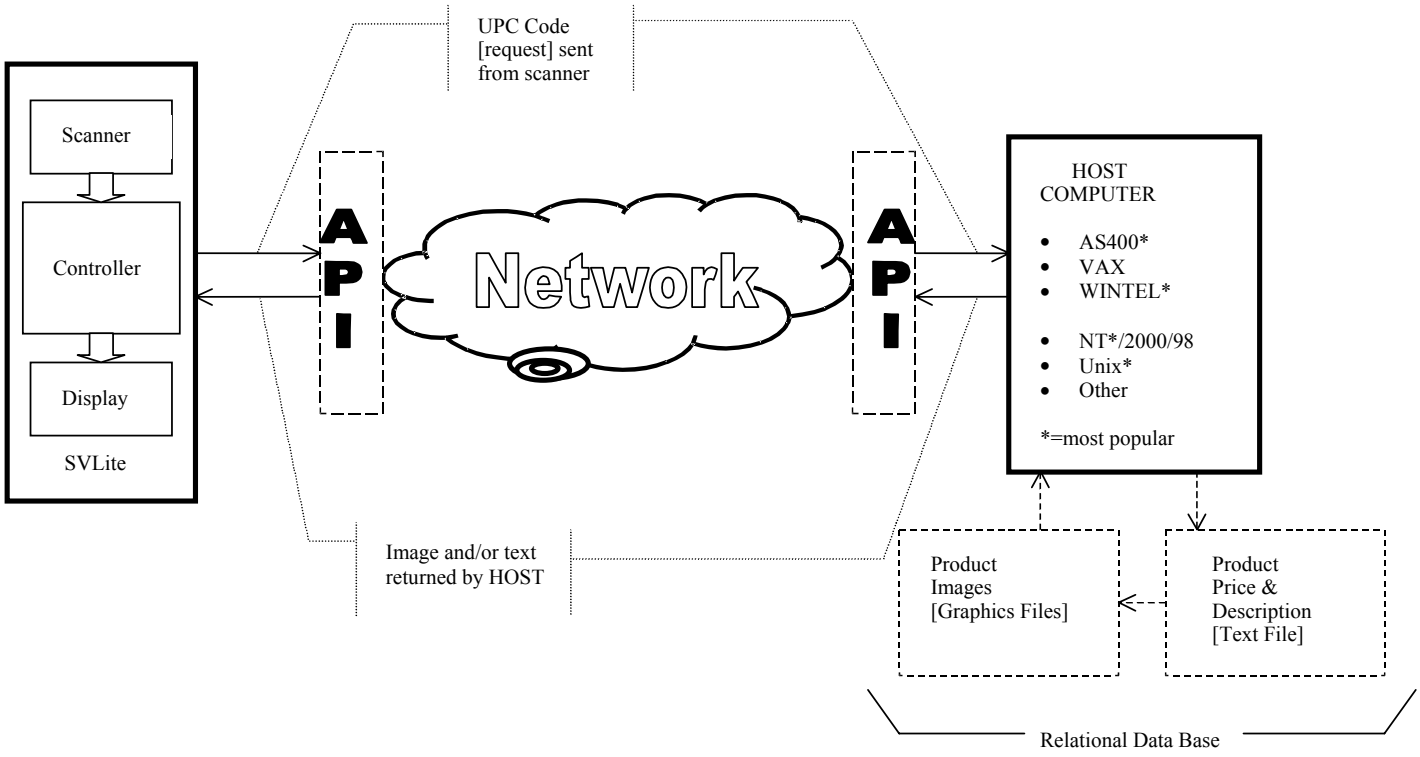
Data Transfer	The unit is not able to send messages to the host even though it has a valid connection. This can be caused by broken server software or for some protocols by firewalls between the unit and host.
File Unavailable	The requested file doesn't exist or security makes it unavailable. This message can display even if everything else is working. It usually indicates a failure to get the .INI or slideshow file. This is generally a recoverable error. For the INI and slideshow files, the unit will get them when they show up.
No Attribute	Indicates that the Windows server has the file, but cannot provide its size or time stamp. This is a failure on the host or server side.

Connect Error

A **Connect** error is the most likely type to occur. The explanations given may not be the only possible reason for the message.

Connect Fail	A general failure to connect to the host. If the host can ping the unit, getting this error should not be possible
Connect Timeout	The initial connection was made but the host has not responded to further communication in timely manner.
Refused	The host is there and the unit can connect to it. The host may not have an appropriate server running, or security may otherwise refuse to allow the connection.
Logon	The user name and/or password are incorrect.
Protocol Fail	For Windows networking, the unit and the host could not find a mutually acceptable protocol level.
Share Unavailable	The directory requested does not exist, or is not shared, or protections prevent the unit from connecting to it.
Host Resolution	The specified host name cannot be resolved to an IP address. This can happen if the host name is incorrect or if the host is on another subnet and there is no WINS server available or the WINS server IP has not been set.
Transfer	This may be a transient error that will not re–occur if you power cycle the unit and try again. Normally you should never see this message, as it indicates a failure of software and/or hardware on the host, the network, or on the unit.

Chapter 5—Interfacing to the Back Office Server



The API (Application Programming Interface) is a description of the ProductInfo protocol that enables the host computer to communicate with ScanVue. It is not a physical piece of hardware. Both ends of the API are physically the same piece of software.

ScanVue only performs 3 functions:

1. Scan the product UPC (barcode) and send it to the HOST computer across the network.
2. Accept the response from the HOST and display the information appropriately
3. Automatically download a slideshow from the HOST and play it.

Fig 8—Simplified SVLite System Diagram

Note:
ScanVue Lite is shipped from the factory configured for DHCP. If your server is located at a corporate location rather than in the store and there are gateways on the network to isolate the stores, DHCP will not be able to set the Gateway IP—it must be set by barcode or by using UnitConfig from a local computer.

Overview

A price verifier specific protocol, 'ProductInfo Protocol', is used to connect price verifier (PV) units such as SVLite to the back office server that stores the price lookup database. A server based application program consists of two major parts—the communication between the PV and the server when a barcoded item is scanned, the data is retrieved and sent back to the PV for display and the part concerned with configuration, control and setting modes of the PV. Both parts are covered in detail in this chapter.

The application program that links SVLite and the back office server database resides on the server. Almost any platform that runs a TCP/IP network can be used as a server. Known hardware platforms in use include AS400, VAX and Wintel. Software platforms include Windows NT, UNIX, Linux, and VMS running Oracle, SQL and other databases.

Software Developers Kit (SDK)

The CD–ROM shipped with SVLite contains the following software:

- ***SetupServer.exe*** program (IEE logo). Creates the necessary directories on the server and installs all required files and utilities into them.
- ***UnitConfig.exe*** install program (IEE logo). Installs the unit configuration program into the \POS directory on the server. UnitConfig is a GUI program used for changing SVLite configuration settings from the server. Written in VisualBasic, the .exe, VB sources and an OCX are provided. When installed, the program icon resembles an open wrench.
- User Guide, Quickstart Guide, test and configuration bar codes and other useful documents. Most documents are in Acrobat (.pdf) format.
- Source code for demo servers and some utilities. The sample programs demonstrate communication with SVLite using the ProductInfo and QFX protocols and include ProductInfo and QFX servers. The VB sources and an OCX are provided for the Unit Configuration program so it may be embedded in the host application. All other programs are written in ANSI C and have been compiled and run without modification under NT, linux, UNIX, and VMS. QFX is a faster, lower overhead alternative to FTP and SMB.
- A number of useful freeware and shareware utilities.
- The monochrome LCD version of SVLite requires images in black and white, ¼ VGA (320x240 pixel), GIF format only. We suggest using **ImageMagick** for image file conversion from other formats into either GIF or uncompressed form. **ImageMagick** is a collection of tools and libraries for reading writing and manipulating images in 68 standard formats including TIFF, JPEG, PNG, PDF, PhotoCD and GIF. Information on **ImageMagic** can be found at www.imagemagick.org.

ProductInfo Protocol Description

SVLite has an embedded TCP/IP protocol stack that is used to send barcodes scanned on the PV to the host computer and receive back the price and description information returned by the servers resident application.

ProductInfo is a TCP based, bi–directional message–passing protocol that uses the same format when moving data in either direction. The SVLite client opens a socket connection for each request generated, usually a scanned barcode, and keeps it open until the server instructs the client to close it. If the server doesn't close the socket, the client will time-out and reset itself. The client can also wait for the server to open the socket thus allowing asynchronous operation.

The protocol can also send an 'event'—usually marking a change of state (opening or closing) of one of the four optional front panel switches. These events may be used by the servers resident application to control functions or logic paths within the application—for instance to change the language messages are displayed in.

An abstract system level diagram showing the relationship between SVLite, the network and the host computer is shown in Figure 8. The API is shown at both ends of the network for clarity. In practice, the application to interface the price lookup tables to SVLite will reside entirely on the host computer.

In the interest of robustness, both ends accept any message whether defined or not; invalid or unknown messages are simply discarded. Such errors terminate the connection. If the client detects it, it may send an error token following re–establishment of the connection in order to log the error on the server. If the server is able to detect this condition, it can log it directly. When the server receives a product query from the PV, it must respond even if the message is just to terminate the connection. Following a query, the client may choose to take an error action if it receives nothing from the server within a defined timeout period. The server can make capability queries and/or mode changes before, during, after, or in lieu of sending any response. If the server wishes to space messages more widely than the client's default timeout, it must send a 'Set Mode' packet to change the timeout; this need only be done once per query, but must be done on each query.

The client may send capability messages regardless of whether the **key name** is known to the server and the server may retain this information. When the server needs to know the value of one of these capabilities, it can consult this retained information. If it is not known, a capability query may be sent and the server may wait a moment for a reply to be received. The reply asynchronously updates the server's information, and the value should be found there by a subsequent lookup following the brief interval required for the client to respond to the query. If it remains undefined, it can be assumed that the client declined to respond, probably because that capability name is not known to it.

Mode settings allow the server to choose between different parameters or behaviors in the client. Theoretically, this can work both ways. If the server wants the client to adopt a certain mode, it sends the command and the client responds. If not, an error report may be generated in response. Mode settings

occupy a separate name–space from the capabilities table. For instance, a mode setting could be used to change the timeout the client uses to decide that a socket connection has broken.

Error reports are used primarily as a debugging tool. The string starts with an error number, optionally followed by white–space and explanatory text. In the nominal case, messages consist of a length, followed by a token, optionally followed by more information as specified by the length and the token. In the trivial case, the message consists solely as a NUL–terminated text string; this is the case when each of the four bytes of length field is an ASCII printable character. When this message type is received by the server it must be interpreted as a product query; it optionally contains the client's identification and white–space preceding the product code. When received by the client, it is interpreted as a single, textual response to a query.

Tokens		Data Types	
Terminate connection	TERM	No Operation	NOP_
Capability query	CAPQ	Display Text	TEXT
Capability	CAPR	Special Text	SPEC
Mode set/query	MDSQ	UNC	UNC_
Mode Response	MODR	URL	URL_
Error Report	ERR_	FIF	FIF_
Product query	PROD		
Data	DATA		
Registration	RGST		
Event	EVNT		

Table 2 Tokens and Data Types

Protocol Implementation Rules

1. Mode values changed during a query session are only kept during that session.
2. The host can make "permanent" changes to mode values for query sessions by connecting to the ProductInfo protocol port (Port 1283) of the client and setting the values. As long as that connection is maintained, the new values will be used in all further queries.
3. Whether a "permanent change" will survive a power cycle of the client is implementation dependent. Clients may provide special functions to record mode information in non–volatile storage.
4. Query sessions are best kept limited to information that is to be displayed immediately, so that additional queries can be answered quickly.
5. Mode values that start or end with white-space must be sent enclosed in double quotes. These quotes are removed when the value is stored. Double quotes within the string are treated as part of the value.
6. When using double-byte font files such as the Kanji font (7640 characters), text strings must be defined as Big-Endian or Little-Endian depending on the server.

SVLite Implementation Rules

1. SVLite's internal ProductInfo server is asynchronous—that means the host may open a connection and send messages to SVLite (the listener) at any time.
2. SVLite sends carriage return/linefeed pairs to indicate an end–of–line (<NUL> marks an end–of–string) to accommodate various host computers but will accept either form.
3. When booted up, SVLite copies the setup information stored in its flash memory into RAM. The copy is the source of mode values when a mode request is executed. Changing a mode's value modifies this RAM copy which is lost when the unit reboots. To permanently save a mode change into flash memory it must be committed and the unit rebooted. A second copy of the setup information is retained as the current value set. This copy is modified by the SCANVUE.INI file contents and other transient settings. This copy is never saved.
4. The mode values read by UnitConfig or modeset are the default values copied from flash when the unit is booted up and may not reflect the current operational values, which is actually the second copy. This situation occurs when the initialization file overrides the default values.
5. SVLite does not always immediately change its behavior when a mode change is sent—most changes require committing and a reboot before they take effect. The modes that can immediately change are: Timeout, TrivialComm, SendUnitID, SendError, SendResponse, TextBackgnd, TextForegnd, and QueryPrefix.
6. Text display modes (TextHPos, TextVPos, TextRows, TextCols) can be forced to take effect without a reboot by setting the TextReset mode to TRUE.
7. The QueryPrefix can contain binary information. To set a QueryPrefix with binary values, use a 'C' language format string for the value. SVLite will accept "\xCC" where CC are 2 hex digits, as a single character with the value of the 2 hex digits; no other sequences are recognized.
8. When booted up, SVLite will attempt to register with the host system. If it cannot do so, it waits 10 seconds and tries again. After each failure, it waits one second longer than the previous time, ad infinitum. Additionally, when SVLite detects that the initialization file has changed, it will make one attempt to re–register. This allows a host computer that has gone down to force all units to register without knowing which or how many PV's are on the network. Registration can be selectively disabled.
9. Text transfers using the default extended ASCII character set shown in Appendix D should be sent in nominal mode. This is because many libraries do not properly handle the characters outside of the usual ASCII range (32–127) correctly.

SVLite Supported Modes

This section defines the device specific implementation for SVLite.

Note: Mode names are case insensitive. Values have case preserved.

Fixed Unit Identification

BuildInfo	The date and time of the software release in the format YYYYMMDDhhmmss; read–only.
ProductName	“SVLite”; read–only.
SerialNumber	Contains the unique serial number for the unit; read–only.

User Configurable Unit identification

UnitID	The host name of the unit, 19 characters maximum.
UnitIP	The IP address of the unit, in standard IP dotted notation.
UnitMask	The network mask for the unit, in standard IP dotted notation.
GatewayIP	The IP address of the gateway machine, in standard IP dotted notation. Required only if access to the various hosts must be routed on the LAN.
UserName	The username SVLite uses when logging into the file server.
Password	The password SVLite uses when logging into the file server.

Setup for Windows Networking (SMB)

Domain	Network domain is required when security is a concern.
DNS	Domain Name Server IP Address (if required to resolve network host names)
WINSServerIP	The IP address of the WINS server, in standard IP dotted notation, if the file server is not on the local network.
WindowsServ	The host name of the Windows server.
ShareName	The name of the shared directory on the Windows server.

Setup for FTP, QFX, NTP, PRODUCTINFO

ServerType	Sets the file host as Windows networking (“SMB”), FTP (“FTP”), or QFX (“QFX”).
FTPServerIP	The IP address of the FTP server in standard IP dotted notation.
QFXPort	The IP port to use for QFX communications.
QFXServer	The IP address of the QFX server.
QFXTimeout	The number of hundredths of a second to wait before timing out on QFX communications.
NTPHost	The IP address of the NTP server to provide the date and time, in standard IP dotted notation. The date obtained is assumed to be GMT.
NTPTZOffset	The number of minutes offset from GMT for the local time zone.
SloppiIP	The IP address of the PRODUCTINFO protocol server, in standard IP dotted notation.

SloppiPort	The IP port to use when sending product information requests and listening for connections.
SloppiTimeout	The maximum wait time, in hundredths of a second, for an initial response to a product information query, kept in non–volatile storage. default=300

Product Query configuration

IgnoreAckNak	Sets SVLite to ignore ACK and NACK characters when received at the beginning of an expected packet. default=FALSE.
IgnoreResponseChars	Sets SVLite to ignore the number of characters specified, from 0–255 at the beginning of the response packet to a query; default=0.
QueryPadTo	Sets the number of text characters for a barcode query to at least the specified number of bytes by adding NUL bytes after the query string. Value is 0 to 64. Has no effect if value <= # bytes in barcode+3. default=0
QueryPrefix	Up to 198 characters. If set, it is prefixed to all query requests.
SendError	Controls whether error messages are sent in response to unknown messages, mode set commands with improper parameters, etc. default=FALSE.
SendResponse	Selects whether confirmation messages are returned after setting a mode's value.
SendUnitID	Selects whether the UnitID string is sent as part of product information requests. Default=TRUE.
TrivialComm	The protocol method to use for product information queries

Presentation configuration

POSTimeout	The time in seconds that price/description text will be displayed before it is cleared and the static display restarted.
ShowShortPoll	The time in seconds between successive attempts to read the INI file if it has never succeeded; default=60
ShowLongPoll	The time in seconds before checking for changes in the INI file. default=300
MsgChecking	Text string displayed when making a product information request. Default string='Checking..Please Wait'.
MsgUnavail	Text string displayed when the product information server does not respond. Default string='Item not Found'.

The following modes are only applicable to the LCD model.

TextCols	The number of columns of text displayed
TextRows	The number of rows of text displayed
TextHPos	The horizontal offset from the left edge where text starts (in pixels). Must be an integral multiple of four pixels. Space must be available on the right side of the line as characters cannot be pushed off the screen.

TextVPos	The vertical offset from the top edge where text starts (in pixels). Must be an integral multiple of four pixels.
TextDisplay	Must be set to EXCLUSIVE for a monochrome LCD or VF text only display.
FontFile	The file name of the current font (only LCD display).

Real Time Clock Display (LCD model only)

ClockFontFile	The filename and path of the font file for the clock display. A different font than the text display may be used.
ClockHPos	Leftmost (horizontal) position of the clock display in pixels. Top left corner of display screen is 0,0, bottom right corner is 319,219
ClockVPos	Topmost (vertical) position of the clock display in pixels. Specifying ClockHPos,ClockVPos defines the top left corner of the clock display.
Clock_Fcolor	Foreground color of clock display (1 or 255)
Clock_Bcolor	Background color of clock display (1 or 255).
ClockFormat	=0; 12 hour display, no AM/PM; =12; 12 hour display with AM/PM
ClockUpdateInterval	=1; displays hh:mm:ss; =60; displays hh:mm =24; 24 hour display.
ScanShowClock	Turns off clock during a scan, or forces it to remain on at all times. TRUE=display clock at all times, FALSE=turn off clock during a scan.

Miscellaneous configuration

DateTime	The current date and time in the format YYYYMMDDhhmmss. This will be correct only if the host has set this value, or if NTPHost is set and SVLite is able to obtain the correct time from the NTP server.
EventEnd	An 8 character string of hex digits (32 bit mask) which enables ending events for a specific device as defined in the table below.
EventStart	An 8 character string of hex digits (32 bit mask) which enables starting events for a specific device defined in table 3 below.

Mode	Value (mask setting)
Disable mask	00000000_h
Pushbutton 0 (leftmost)	00000001_h
Pushbutton 1 (2^{nd} from left)	00000002_h
Pushbutton 2 (3^{rd} from left)	00000004_h
Pushbutton 3 (rightmost)	00000008_h

Table 3 Event Start (pushbutton) Masks

	Start and end masks are both present in a transmitted packet but one is always disabled. An input device using both start and stop events will generate two complete packets, one for start and one for stop. For most applications the end mask is not required and should be disabled. Examples and a full explanation of Event mask usage is in the *Using Input Devices* section on page 33.
User1	A string up to 255 characters long that can be set and read by the application.
RegisterINI	SVLite will register itself with the ProductInfo host after successfully reading a new INI file.
RegisterStart	SVLite will register with the ProductInfo host upon startup.
DisplaySetup	Display the first information screen at startup.
DisplaySetup2	Display the second information screen at startup
DisplayAll	Display passwords on the information screens
NoBarCodes	Disable configuration by barcode
Version	Returns a string containing the software version number
FlashHash	A CRC calculated on the content of flash memory. The CRC is displayed on the initial bootup status screen. Display format is CCCC–TTTT where the 1st four hex digits are the calculated CRC and the 2nd four hex digits are the received CRC from the host. Both CRC's must match to ensure no file corruption during download. There is no error correcting mechanism at this time. Also displays in UnitConfig.
Port2Function	Changes the function of the external RS232 port. Port parameters are fixed for each device (Value). The port passes data transparently in both directions. The port function can be changed with modeset or UnitConfig utility. Factory default is scanner mode.

Value	Function
SCANNER	Receives data from hand scanner (input port)
OUTPUT	Passes through data sent by server (output port)
BIDIRECTIONAL	Operates in both directions
KEYPAD	20 key keypad

Port2Input — Defines whether Scanvue sends a character at a time or a complete line to server. default value=CHARACTER

Value	Function
LINE	Buffers all characters until CR or LF eceived then sends entire string to host. Typically used with MSR
CHARACTER	Sends each character as received to host. Typically used with NCR K590 printer

BypassNCRPrefix

Instructs Scanvue to accept NCR prefixed bar codes for programming. default value = TRUE. NCR prefix mode does

not have to be turned off to be able to program the unit with barcodes.

ExtCharacterSet

Applicable only to VFD display models. Selects the character set (language) for display. The default character set contains 'European' in 64 positions AO-DF. The extended language character sets are shown in Appendix D.

Character Set	Value
ASCII+European	EUROPEAN (default)
ASCII+Katakana	KATAKANA
ASCII+Cyrillic	CYRILLIC
ASCII+Hebrew	HEBREW

Table 4 VFD Display Language Selection

PLUFile A price-description lookup file specified in the [display] section of scanvue.ini that is loaded into the unit at boot time and checked for updates every 5 minutes (default value). The PLUFile name cannot be changed from UnitConfig.

SlideShowFile The name of the slideshow file specified in UnitConfig. Cannot be changed from UnitConfig.

Command modes

Commit Set TRUE, causes the mode settings in the RAM copy to be written into flash and become the defaults for future operation.

Invalidate Sets all mode values to factory defaults and restarts the unit.

Restart Set TRUE, causes the unit to restart. This must be done after a commit to permanently save any new settings into flash.

TextReset Set TRUE, causes the settings for displayed text to be immediately executed. For the text values to become permanent, the commit and restart commands must also be issued.

Programmatic Modes

Programmatic modes are for use within another program and cannot be read or set interactively with UnitConfig or modeset. For example, UnitStatus is a mode used within the 'heartbeat' utility program to report on the status of the unit being read. These modes may also be used within your own program

GiveHelpText Appends help text to null MDSQ requests.

Timeout The maximum wait time, in hundredths of a second, for an initial response to a product information query. Valid for the current connection only.

UnitStatus	Reports on the state and activity of the unit. Returns a single line of values separated by spaces. Counters restart from zero if maximum count exceeded or unit is reset. Mostly applicable to LCD units only. In order the values are: Number of seconds unit has been running since last reset. Largest number is 42949672 (~16 months). A single character indicates the current state of the slideshow. N=no slideshow loaded; R=slideshow running; L=slideshow being loaded. Total number of slideshows that have been successfully loaded and started. Maximum count 65535 Number of slides defined in current slideshow script. Maximum number 65535. If not running, it is the number in previous show or the current show that was not able to load slides. Number of slides successfully loaded. If the show is running this is the actual number of running slides. If not running, it is the number for the previous show or the current show that was not able to load slides. e.g., 160 slides defined, 147 loaded. Total number of slides displayed since last reset. Maximum value is 4294967295. Total number of images displayed since last reset. Maximum value is 4294967295 and includes text messages, start up screen, frames from pushed animations… Number of barcodes scanned since startup. Maximum value is 4294967295 and excludes barcodes scanned while the unit is busy with another product request. Number of responses from the host to product info queries. Maximum value is 4294967295. Includes both positive and negative responses. Number of times the unit received no response from the server. Maximum value is 4294967295.

QFX Quick File Transfer Protocol

QFX is a very low overhead protocol that is a much simpler and faster alternative to FTP for moving files quickly across the network. There is no authentication. The QFX server allows only read access to files in and below its default directory; this is the extent of security.

QFX packets consist of a 4–byte (network order) packet length, a 4–character token, and possibly data. Packet length includes the 4–byte length field.

Tokens are:

INFO	File information request/response
SEND	File send request/response
DIFF	File timestamp comparison
ERRR	Error response

INFO exchange:

request: [length][INFO][filespec <NUL>]

response: [length][INFO][YYYYMMDDhhmmss size <NUL>]
Four digits for year, two each for month (January is 01), day (01–31), hour (00–23), minute (00–59), and second (00–59), followed by a single space–character, and finally the size of the file, in bytes.

response: [length][ERRR]
Requested file does not exist or request is malformed.
Explanatory message is optional, should be string consisting of number, whitespace, optional text.

request: [length][SEND][filespec <NUL>]

response: [length][SEND][file data]

response: [length][ERRR]
Requested file does not exist or request is malformed.
Explanatory message is optional, should be string consisting of number, whitespace, optional text.

request: [length][DIFF][filespec <NUL>][YYYYMMDDhhmmss<NUL>]

response: [length][DIFF][Bool]
Bool value is TRUE if the file's date/time are DIFFERENT than the provided timestamp and FALSE if it is the same.

response: [length][ERRR]
Requested file does not exist or request is malformed.
Explanatory message is optional, should be string consisting of number, whitespace, optional text.

[filespec] uses UNIX style forward–slash directory notation. It is the responsibility of the server to convert the notation to the native method, and to guarantee that the path cannot extend outside of the default directory.

The directory '/' refers to the current directory; a leading '/' is ignored. The QFX server must guarantee the requested file is in or below its current directory.

Line–terminators are newline characters; carriage returns are optional.

The server is responsible to read text files and transmit them with appropriate line breaks. Specifically, this applies to VAX/VMS.

Using Graphics Characters (LCD Display)

SVLite with an LCD display can display graphics characters such as Kanji and Chinese (ideograms) as well as standard Latin text characters. The graphics font file is downloaded into the price verifier at boot up time by a command line in the scanvue.ini file. As only one font file can be stored at a time, the characters displayed are limited to those in the currently loaded font set. Certain font files contain other characters; for example the Kanji font set also contains English, Greek and Cyrillic characters, and GB (simplified) Chinese includes 128 US-ASCII characters. Note: although standard US-ASCII characters are single byte codes, using the US-ASCII characters in the GB Chinese font file requires the double byte codes specified in that font file.

Displaying graphics characters requires two hex bytes (double-byte text) per character due to the large number of glyphs in the font set (Kanji has 7,640 characters). The rules for using double byte text are shown here.

Double byte text strings are formatted in one of two ways which determine the byte order, Big-Endian or Little-Endian. The server hardware determines if you will need big endian or little endian byte strings. Double byte strings sent in trivial mode must be double NUL (00 00) terminated.

Big-Endian text strings start with the bytes FE FF and end with the bytes 00 00.
Ex: **FE FF** b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}.....**00 00**

Little-Endian text strings start with the bytes FF FE and end with the bytes 00 00
Ex: **FF FE** b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}.....**00 00**
where b_1-b_{10} are a string of 5 double-byte character codes.

In scanvue.ini, double byte strings can be used for wallpaper but the text must start with << instead of < or // instead of /.

For the Demoserve data file (data.dat) on the CD-ROM, the characters |> indicate a Big-Endian string and |< indicates a Little-Endian string. Note: Demoserve actually sends the FE FF or FF FE bytes not |> or |<.

A utility program, dmp2gdf, provided on the CD-ROM creates a .gdf (double-byte) font file from a single byte ASCII font file.

Using Input Devices

SVLite has two optional input devices—a hand held scanner that sends its data exactly like the internal scanner and four front panel mounted switches that send data by a special message packet containing an event (EVNT) token.

Hand Barcode Scanner

The hand scanner performs the same function as the fixed scanner but is attached to SVLIte by an umbilical cable so the scanner can be moved to the item rather than physically moving the item (which may be bulky or heavy) under the fixed scanner.

Push Buttons

Four pushbutton switches are located on the front bezel directly under the display. Pushing a button generates a start event packet and releasing it generates an end event packet. Event packets sent to the host contain the switch number activated (1 is leftmost switch and 4 is rightmost switch) thus the host computer can make decisions based on the particular switch pushed.

The time the button is pressed and held down can be measured by setting the start and end event masks and reading the number of 10 millisec clock ticks between them. For example, this could be useful in an application that uses the difference between a ‘short’ press and a ‘long’ press.

- Set start mask. Set end mask only if required for timing

Start Event and End Event Masks

Start and end event masks are modes that can be setup with UnitConfig. The values consist of 8 hex digit (32 bit) masks that define the type of device initiating the event as shown in the table below. Factory default is all masks disabled (00000000_h). Start and end masks are always present in the transmitted packet but one is always disabled. For most applications the end event mask is not required and it should be left disabled. A start event is a complete packet and an end event is a complete packet. Set the mode values (mask settings) as shown in Table 1.

Mode (Description)	Value (Mask Setting)
Disable mask (factory default)	00000000_h
Push button 0 (leftmost)	00000001_h
Push button 1 (2^{nd} from left)	00000002_h
Push button 2 (3^{rd} from left)	00000004_h
Push button 4 (rightmost)	00000008_h

Table 5 Pushbutton Masks

Structure of EVENT packet

1. The EVENT packet format for push buttons is described earlier in Chapter 5 in the "SVLite Supported Modes" section but is repeated here for completeness.
2. Separate packets will be sent for start events and end events (if used).

Examples (shown in hex form)

Push button start event packet for button 2 (no unit ID)

[00000021 45564E54 00000004 00000000 00000000 00000000 00000000 00000000 00]$_h$

length	EVNT	start mask for button 2	stop mask	tick count	not used	not used	not used	terminator

Pushbutton end event packet for button 2, 500mS button press, (no unit ID)

[00000021 45564E54 00000000 00000004 000001F4 00000000 00000000 00000000 00]$_h$

length	EVNT	start mask	stop mask for button 2	tick count 500 mS	not used	not used	not used	terminator

SVLite420 Vacuum Fluorescent Display

Note: references in this manual to the use of graphics images, double-byte text and downloaded fonts are not applicable to the SVLite 420 VFD model.

Description

SVlite420 has a 4 line x 20 character x 5mm high, 5x7 dot matrix vacuum fluorescent (VF) display and shows text in a fixed font size and pitch. The default character set is US-ASCII extended with 64 European characters. There are three (3) more 64 character extended language sets Cyrillic, Hebrew and Katakana that can be selected with the UnitConfig program by entering the character set name in the mode **ExtCharacterSet**. This feature allows the end user to set the character set to the language of choice. Once SVLite420 is set to display one of these optional character sets, any message sent to the display will use the characters in that chosen language set. The language character sets are shown in Appendix D and always contains the US-ASCII characters shown in Table 4 plus one of the optional 64 character sets shown in Tables 5-8.

The user may change any or all of the 3 internal customer messages (factory default is English) into one of the 4 language character sets available on the VF display. This will allow the user to display customer messages in the language of the country the unit is installed in.

As this model cannot show graphics or inherently run a slideshow, the customer greeting screen shown in Figure 15 is normally displayed as a static screen. It is possible to create a loop of linked text messages on the server side that can be sent to SVLite420 in a timed sequence to create a 'text slideshow'.

Operation

When SVLite420 finishes booting up it will display the info status screens first, then the logo screen and finally the static greeting screen shown in Figure 15. This screen will be continuously displayed until either an item is scanned or the host server sends a different message.

```
Welcome to
IEE SVLite420
Scan Your
Item Here
```

Fig 9—Greeting Screen

When a barcoded item is read by the scanner, SVLite sends the ASCII characters to the server using the protocols described in Chapter 4 and 5. The mode SendUnitID will transmit the units ID as well as the ASCII

characters when set TRUE (default). If the VF unit is mixed with LCD units in the same installation, it is important to keep this mode set true so the server application can discriminate between VF and LCD unit requests and not send graphics images to VF units. A VF unit receiving an image will display the default no response message then will display a blank screen. The host server should respond to a VF unit with a formatted text information block (TIB) of all 80 characters, including the blanks, to locate the actual text in the required positions on the lines (Figure 16)—there are no direct text formatting commands in the present version.

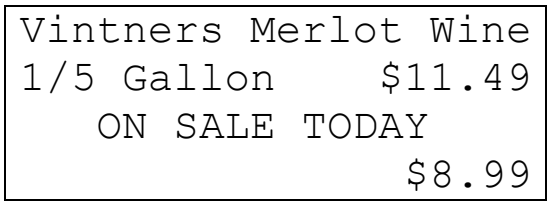


Fig 10—Item Description & Price Display

Text Slideshow

If you want to send multiple messages in sequence, this must be controlled from the host server end. For instance, a text based series of messages for continuous advertising could be sent to the display under control of the host. The host would have to have the sequence of messages in a message file (or multiple small files) and send them in a sequential loop with a delay for adequate viewing time. The loop control would require an interrupt when the host receives data from the ScanVue420 barcode reader to stop the message sequence, retrieve the item description and price, display it for a set time and then return to the message loop.

Changing Language Character Sets

Appendix D shows the VF displays' default ASCII font set with 64 empty character codes between A0 and DF and the four extended language character sets that can fill these empty positions. Initially, the default character set is US-ASCII+European. The ASCII characters are always available but the extended character set can be changed from UnitConfig by selecting the mode ExtCharacterSet and entering the appropriate value in the New Content box. Only one extended language at a time can be loaded.

Extended Language	Value
ASCII+European	EUROPEAN (default)
ASCII+Katakana	KATAKANA
ASCII+Cyrillic	CYRILLIC
ASCII+Hebrew	HEBREW

Table 6 VFD Display Selecting Language

Customer Messages

Three built in customer messages are available in SVLite. These messages provide specific information to a customer using the price verifier. SVLite is shipped from the factory with preset default messages. The message content (text) and language may be changed by the user for use in other countries. The 3 default messages are:

Checking…...One moment please this message is displayed when an item has been scanned and there is a delay in returning the price and description—usually due to heavy network traffic.

Unavailable…..please try later this message is displayed if SVLite420 times-out before the server returns price and description or an error message. Usually means the network is down but initially might mean the item cannot be found in the price lookup table because it is coded incorrectly.

Welcome to IEE SVLite420….Scan your item here Greeting screen (see Fig 15). The static 'greeting screen' that is displayed to a customer.

Changing Customer Messages

1. Change the character set to the one you want to use from UnitConfig on the **ExtCharacterSet** line. Make all the changes before restarting. The default character set is US-ASCII+European.

2. When creating new messages in UnitConfig, standard characters (that is, those available on your computer keyboard) can be entered directly from the keyboard. Non-standard characters must be entered in hex code form. Hex codes consist of 4 hex digits (\xdd; where \x specifies that a hex code follows, and dd is the hex code for the character you want as shown the character code table) but occupy only 1 character space on the display. See examples in Appendix H–Applications Notes.

3. **Checking…** and **Unavailable…** messages can be changed by entering the new text on the ***msgChecking*** and ***msgUnavailable*** lines in the UnitConfig program screen surrounded by quotes (i.e. "message"). Non-standard ASCII characters must be entered as hex codes as described above. If spaces are required, for instance to center the text, all leading spaces must be entered.

4. The **Welcome to IEE…** greeting screen can also be changed in UnitConfig. Enter the new text on the ***BannerText*** line surrounded by quotes. Up to 80 characters can be entered on the line with spaces inserted at the appropriate places to center or offset text on each line of characters. When the message is sent to the display, it will word wrap at the end of each 20 character line.

For Example:

```
    Welcome to
  Jims Hardware
    Scan Your
    Item Here
```

The text file string to create this message is shown below. In this example * is used to denote a space. This example is only shown this way to make it clear, and is not required when you create an actual file.

"*****Welcome*to********Jims*Hardware********Scan*Your**********Item*Here******"

There is a real example of changing messages and language in the Application Notes section (Appendix G).

5. After changing the messages, click the **Set Mode**, **Commit** and **Reboot** buttons in sequence. The unit will reboot and the new messages will be saved and active once the unit is running again.

Appendix A—Configuring with Barcodes

Barcode Reader Settings

The Intermec E1022 barcode reader is a independent device from ScanVue containing its own processor and firmware. The scanner is configured with special barcodes created by the **Intermec EasySet program**—for instance, different symbologies can be programmed into the scanner. A copy of this program can be downloaded from Intermec's website at www.intermec.com and is also available in the "Third Party Utilities" directory on the CD-ROM.

A different set of barcodes is used to configure ScanVue network settings. The actual printed barcodes are in the documentation folder on the CD-ROM in a file called barcodes.pdf. The codes are also shown in the table on page 39-40 if you want to generate additional codes. These codes should be in made in code 128

Recommended Scanner Settings

The following barcode tags should be scanned to set the barcode reader into the correct mode for use with SVLite. An SVLite unit shipped from the factory will already be configured this way.

reset factory defaults

\46\42\60

autostand mode - active for activation cycle duration (10 s), repeat activation cycle after good rea

\4C\40\01\47\50\46\4D\47\51\0A\60

RS-232 - baud rate - 9600 (*)

\41\07\60

UPC / EAN - re-encoding UPC-A, UPC-E, EAN-8 - UPC-A transmitted as UPC-A

\44\4E\60

Code 128 / EAN 128 - active

\41\5A\60

Configuring SVLite®

If you are not connected to a network SVLite can be configured for the network (IP Address, Gateway……) by scanning barcodes instead of sending the commands over the network from UnitConfig. Barcode should be created in code128 symbology.

Create barcode tags containing the appropriate network information as shown in Table 2, pages 49 and 50. A program such as **'Avery Label Pro'** or **'B–Coder Lite'** from Taltech can create barcode tags for you. The scanner, which is a separate device from ScanVue must also be configured correctly to read the ScanVue tags. Scanner configuration tags are created with Intermec's EasySet program. The scanner will beep once when it reads correctly. If multiple rapid beeps occur, the code probably didn't read correctly. If in doubt, scan the 5 tags shown on the previous page to put the scanner in the correct mode.

Follow the step–by–step procedure outlined in the following pages to configure SVLite® using barcodes:

1. Power up the SVLite unit. Wait until it finishes booting, the 2 status screens have been displayed and the IEE logo clears.

2. Scan the following ScanVue barcode tag.
   - **Recall Defaults**

   Each label will generate a single beep when scanned properly, then a double beep to acknowledge the operation is complete.

3. Scan the specific labels (listed in Table 1) required to change the configuration to your network's parameters. Each setup label will beep once when scanned correctly. SVLite will display the raw code, then the setup name and entered value. Frequently used bar codes are provided as a .PDF file on the CD–ROM and may be printed and used directly from the page.

4. When all the setup labels have been scanned in and visually verified, scan the **Save Configuration** label. This causes the complete configuration setup to be written into flash memory and reboot so the changes will take effect.

5. At this point, use the Intermec EasySet program to generate and scan in the configuration for the barcode scanner. Recommended settings are above. The most common configuration requires scanning the **Enable NCR Prefix ID** code. This prevents any legitimate UPC barcode from being read as a configuration barcode.

6. This completes the network configuration for SVLite. If your network servers have already been set up, SVLite will start communication with the network normally.

Mode/Setting	Barcode	Description	Default Setting
Unit ID	**IDL**SS……S	A unique unit name, 19 characters maximum.	SVLITE®
Unit IP[1,4]	**UI**AAAAAAAA	IP Address. Each unit must have a unique	10.0.0.227 or DHCP
Network Mask[1,4]	**UM**AAAAAAAA	IP Sub–net mask	255.255.0.0 or DHCP
Windows Server[2,7]	**WSL**SS……..S	Host name of the file server, 83 characters max.	SVSERVE
Shared Directory[2]	**SDL**SS……...S	Share name on the file server, 83 characters max.	POS
File Server IP[1,7]	**SF**AAAAAAAA	IP Address of the FTP–based file server.	Not Set
Network User Name[2]	**WUL**SS……..S	Used to connect to the file server, 19 characters	GUEST
Network Password[2]	**WPL**SS……..S	Used to connect to the file server, 31 characters	Not Set
Network Name[2]	**NNL**SS……S	ESS ID–network name for wireless networks	IEE
DNS IP[1]	**DN**AAAAAAAA	IP Address of DNS	Not Set
Gateway IP[1]	**GW**AAAAAAAA	IP Address for routed or segmented networks.	Not Set
WINS Server IP[1]	**WW**AAAAAAAA	Required for Windows networking.	Not Set
Domain/Workgroup[2]	**DWL**SS……S	Domain or workgroup name, 83 characters max.	IEE
Product Info IP[1]	**DS**AAAAAAAA	IP address of users ProductInfo server.	10.0.10.13
Product Info Port[3]	**DP**nnnn	Port number for ProductInfo.	1283
Product Info Timeout[3]	**TO**nnnn	Duration (secs) product information is shown	30
Wireless RF[3]	**WN**0001	Selects wireless RF mode	Wireless=T, otherwise=F
10baseT Ethernet[3]	**WN**0000	Selects hard wired Ethernet	Hardwired=T,
Save Configuration	**KQ**0000	Save configuration in EEPROM.BIN file & reboot.	N/A
Backup Configuration	**CFGSAVE**	Create backup of EEPROM.BIN file	N/A

Mode/Setting	Barcode	Description	Default Setting
Restore Configuration	**CFGREST**	Restore backup to EEPROM.BIN file	N/A
Factory defaults	**IN**123456789	Scan twice; restores factory default configuration	N/A
Reboot unit	**IN**987654321	Scan twice; reboot without saving config	N/A
Scanner defaults	EasySet Program	Sets Intermec scanner to its default settings	N/A
Show config screen 1	**IS**0001	Displays 1st config status screen	N/A
Show config screen 2	**IS**0002	Displays 2nd config status screen	N/A
Reboot Immediate	**RS**0000	Immediately reboots unit	N/A

Table 7 —Barcode Configuration Labels

[1] IP addresses ('XXXXXXXX') are in hexl notation. Each of the quads in the address becomes a pair of hex digits (e.g., 10.0.0.10 is encoded as 0A00000A).

[2] String values start with the count of characters ('L') followed by the characters of the string ('SSSSSSS'). SVLite® preserves the case of characters in strings even if it is not meaningful to the network or host. If the string starts with a digit, the length field must be 2 digits i.e., 01, if the string starts with a non–digit the length field can be 1 or 2 digits.

[3] Numerical values ('L' and 'NNNQ') are decimal. Numerical values must be followed by a non–digit character. Purely numerical parameters (e.g., Product Info Port) should be followed by an upper–case alpha character to prevent confusion caused by check–characters and stop codes.

[4] If shipped as a hardwired unit it has the default IP address shown. If shipped as an RF wireless unit the default is DHCP.

[5] The barcode is shown in the Metrologic user manual

[6] There are 2 different screens depending on the type of RF radio card installed.

[7] Setting the Host name also sets SMB file–access mode. Setting the File Server IP also sets FTP file–access modes.

Command barcodes must be at least six characters long, including the two–character prefix but excluding the check character and any start and stop characters. Extra padding characters may be added to guarantee this. Padding characters should be upper case alpha characters and are ignored. Numerical values may be zero padded (e.g., '0001' instead of '1').

It is highly recommended that barcodes be created in Code 128. The barcodes shown in the table are in Code 128 format. Some other codes generate numeric check characters which MUST be terminated with an alpha character to prevent ambiguity.

The following characters or combinations must NOT be used: A, F, E0, FF, B1, B2, B3

Appendix B—SVLite Initialization File

Overview

A default **scanvue.ini** initialization file is placed into the share directory on the server by the installation program when the server software provided on the CD–ROM is installed. The **.ini** file must be on the server, in the share directory and the SVLite units which are hard coded to look for the **.ini** file in the share directory, must find it or they will not be able to establish a connection to the network. The **scanvue.ini** file is read by every price verifier on that network server once a communication link is established. The **.ini** file controls the behavior of every unit linked to that server.

If you have the LCD model, scanvue.ini defines the text display characteristics including font type, number of rows and columns and text position within the display viewing area.

Real Time Clock

A real-time clock display can be superimposed on the LCD screen (see Fig 17). The characteristics of the clock display are defined in the scanvue.ini file. SVLite maintains an internal clock tick counter but does not store 'real time'. Date and time are initially set by the NTPServer and TZOffset modes in the.ini file (see example p49) or UnitConfig and should be periodically updated by the DateTime mode from your server application. The host must be running an NTP or similar time server that the SVLite application can access for the current date and time.

The clock display font can be defined independently of the font used for text display—this is the only case where 2 different fonts can be used on screen at the same time. The clock feature is intended for the stores internal purposes rather than for customers and may be used for example, as a time clock for employees to clock in and out of the store by scanning their ID cards under the barcode reader.

Fig 11—Real Time Clock Display (LCD only)

Rules for the scanvue.ini File

1. The **.ini** file is a text file consisting of sections specifying different characteristics of the price verifier. Each section is marked in enclosing square brackets i.e. [Group].
2. Sections may be in any order. Initialization lines may be in any order within a section.
3. Lines that start with a single quote (‘) or octothorpe (#) are comments. Blank lines and white space (except possibly in a value field) are ignored.
4. Initialization lines consist of a token followed by blanks or tabs, an equal sign, more blanks or tabs, followed by the initialization value for the token. i.e. TOKEN=Value or TOKEN = value are both valid forms.
5. Tokens/Value pairs may be directed to an individual unit by preceding the token with the Unit ID enclosed in angle brackets i.e. <Tools1> Text_Rows = 4
6. Tokens/Value pairs may be directed to all units in a group by preceding the token with the Group name enclosed in angle brackets i.e., <Shoedept> Text_Fcolor = Green.
7. Group names may only consist of alpha and numeric characters.
8. Tokens and sections that are unrecognized are ignored. Default values are used when a token is not present.
9. Values are set in the order they appear in the file. If there are duplicate tokens in a section, the value used is the last one on the list except when a Unit ID is specified. If the Unit ID is specified, only that token/value pair will be used, regardless of the position within the section.
10. A font file must be specified (replaces the default font) if the text foreground and background colors are to be specified.
11. If the clock function is being used, a second font file may be specified for the clock display. This is the only case where 2 font files can be used at the same time.
12. The NPTServer mode sets the time when scanvue.ini is run (at boot time). To keep the clock accurate, periodic updates using the TimeDate mode from your application is recommended.

Sections & Commands

(note: commands are not case sensitive)

[DISPLAY]	Configures the way text is shown on the LCD display. Only applicable to the LCD display.
Text_Rows	Number of rows in line–display operation mode.
Text_Cols	Number of columns or characters in line–display operation mode.
TextHPos	The horizontal offset from the left edge where text starts (in pixels). Must be an integral multiple of four pixels. Space must be available on the right side of the line–characters cannot be pushed off the screen.
TextVPos	The vertical offset from the top edge where text starts (in pixels). Must be an integral multiple of four pixels.
FontFile	The full path for the file that contains the font.

SlideshowFile	The full path for the slideshow file.
PLUFile	The full path for the PLUFile (lookup file)
[Display]	The following commands are also in the [Display] group but are usable on both LCD and VF display models.
Msgchecking**	A text string displayed when a barcode is scanned. The default string is ***Checking…Please wait***. When the host returns the price and description, it overwrites this message. Alternatively, the value may be an image filename (*/filename*) instead of the message string, in which case the image (wallpaper) will be displayed.
MsgUnavail	A text string displayed when a barcode is scanned and the host cannot be reached. The default string is ***Unavailable…Please try later***.
POSmOut	The number of seconds that text will be displayed before it is cleared and the slide show (or Welcome message in the case of a VF display) restarted. Can only be used in the .ini file. Performs the same function as the POSTimeout mode (in UnitConfig or modeset).

*A font file must be specified (to replace the default font) to use these two parameters. The 256 color palette can be displayed on a desktop computer by running the file [ScanvueTextColors2.htm) on the CD_ROM from from a browser.
**Msgchecking can specify either a message string or an image file (called wallpaper) but not both.

[GROUP]	Assigns individual price verifier units to a group.
GroupName	The Token is the name of the group and the value is a list of one or more Unit IDs, separated by commas that belong to that group. The same group name can appear on more than one line, with more Unit IDs listed. Units may belong to more than one group. Note: if both LCD and SVLite units are used in the same installation , the VF models should be placed in a different group than the LCD models to prevent graphics images being sent to them as they do not respond to images and will display a blank screen after the “unavailable….try later” message.
[REDIRECT]	Allows a specified unit to be redirected to different initialization file
UnitID	The Token is the unit’s ID number, and the value is the UNC for the new initialization file. If SVLite recognizes its ID in the list of IDs, it uses the value to obtain the new .ini file. A list of which initialization files have been referenced in this process must be kept so that loops can be detected and terminated.
[CLOCK]	Superimposes a real time clock display on the LCD screen
ClockFontFile	The full path for the file that contains the required font. This is the only case where a second font file can be specified to be run.

ClockTransparent	If set to transparent, there is no text box around the display. Value=TRUE, clock text box is transparent. Value=FALSE, clock is in a text box of background color.
ClockHPos	Defines the horizontal (column) position of the left side of the clock display box (in pixels). The left column pixel is 0 and the right column pixel is 319
ClockVPos	Defines the vertical (row) position of the top of the clock display box (in pixels). The top row pixel is 0 and the bottom row pixel is 219
ClockFormat	Defines the format of the clock display. Value=0, 12 hour display without AM/PM. Value=12, 12 hour display with AM/PM. Value=24, 24 hour display.
ClockUpdateInterval	Defines the clock display update interval in seconds. If a multiple of 60 the seconds display is suppressed. Value=1 to 59 inclusive, displays count of seconds. Value=60, seconds display is suppressed.
ScanShowClock	Defines if clock display is suppressed during a scan cycle. Value=TRUE, clock display is on all the time. Value=FALSE, clock display is suppressed during a scan cycle.
NPTServer	Gets the time from a time server. Value=<IP Address>, IPaddress of the time server
TZOffset	Defines local time zone offset in minutes from GMT. Value=n, n=# minutes difference from Greenwich Mean Time. US East Coast offset is 300 (minutes) from GMT

Example of Initialization File

 # Define which SVLite units are grouped together. TextOnly group are
 # VF display units (fixed size fonts) while Fourline and Twoline groups are
 # LCD display units set up for text use with specified font. PLU file shown.

```
[GROUP]
ToolDept      =tools1, tools2, toolsentrance
Cosmetics     =cosmet1, cosmet2, makeup1
HouseWare     =kitchen, china, bridal, appliance
FourLine      =tools1, tools2, cosmet1, kitchen
TwoLine       =toolsentrance, cosmet2, makeup1, china, bridal, appliance
TextOnly      =stationery, tirecenter, garden
```

 # Set up the display characteristics

```
[DISPLAY]
<FourLine>    Text_Rows    =4
<TwoLine>     Text_Rows    =2
              Text_Cols    =20
<FourLine>    FontFile     =VGAfonts/cour4x20
<TwoLine>     FontFile     =VGAfonts/cour2x20
Msgchecking   = Checking….Please Wait
```

```
<Tooldept>     SlideshowFile =shows/tools.sho
<HouseWare>SlideshowFile =shows/housewares.sho
                SlideshowFile =shows/everybodyelse.sho

<PLUFile = c:\POS\lookup.txt
<SlideShowFile> = c:\POS\Ferrarishow
```

 # Members of these groups are directed to a different .ini file

```
[REDIRECT]
<HouseWare>InitFile          =tools.ini
<FourLine>     InitFile          =textonly.ini
```

#Put a real time clock display in the bottom left corner of the screen
#Clock feature only available on SVLite LCD models

```
[Clock]
#Clock font is 8x14 pixels/char
ClockFontFile=C:\POS\fonts\bit8x14.gft

#Clock position is bottom left corner of screen
ClockHPos=0
ClockVPos=300

#Foreground (character) and background (display box) colors
Clock_Fcolor=255      #white
Clock_Bcolor=0        #blue
#Make the display box disappear so the characters are floating on the image
ClockTransparent=TRUE

#12 hour clock display with AM/PM meridian
Clockformat=12

#Display seconds
ClockUpdateInterval=1

#Turn off clock display during a scan cycle
ScanShowClock=FALSE

#Get time and date from time server and send to Scanvue
NPTServer=10.12.14.16

#Pacific coast time is 8 hours (480 minutes) later than GMT
TZOffset=480
```

Appendix C—Creating a Slideshow

This section contains detailed instructions for creating a slideshow file. The easiest way to make a slideshow is to use a text editor like Windows® Notepad to create the script necessary to run the graphics images (slides).

A **slideshow** is a file containing graphic images and a script (the instructions) on how these images are to be displayed. The number of images in a slideshow file cannot exceed 30. Slideshow files reside on the network in the servers' share directory and are 'pushed' to the displays by commands in the slideshow file and stored in the unit's RAM memory. The slideshow is executed from within the unit.

A **slideshow** file consists of a sequential list of references to graphics images in the order in which they are to be displayed, specifying the duration each image is displayed. The file uses Token=Value pairs to define the sources of the image files. Blank lines, lines starting with a single quote (') or an octothorpe (#) are ignored, as is white space around tokens or the equal sign separating the token and its value.

Creating a Slideshow File

The **SLIDESHOW** section is a script that is interpreted by SVLite as it is executed. Slideshow entries consist of two items of information: the slide image by sequence number and the duration the image is to be displayed in ticks where a tick=1/100 second. A **FRAME** may be defined to equal a number of ticks; for example FRAME=100 makes FRAME = 1 second. Frame can be shortened to F, which can then be used as the duration (F1 = 1 second or F3 = 3 seconds)

CAUTION: DO NOT DISPLAY A FIXED IMAGE FOR LONG PERIODS OF TIME AS THIS MAY CAUSE A PERMANENT LATENT (GHOST) IMAGE ON THE LCD. THE LCD WARRANTY DOES NOT COVER THIS EFFECT.

This effect known as "imaging sticking" is caused by an ion charge buildup in the liquid crystal material and is inherent to all LCD displays. The LCD manufacturer recommends no longer than 5 minutes before changing the image, the purpose being to turn 'off' the pixels that are 'on' for a period of time. This will prevent image sticking from occurring.

Appendix D—Font and Language Sets

Graphics (LCD) Display Models

SVLite with LCD display has the built–in default font set shown in Table 3. The 16x32 pixel font has the extended ASCII character set (0–255) and can be displayed 20 characters per line. A different font can be downloaded when the unit boots up by specifying it in the scanvue.ini file. Only one font at a time can reside in ScanVue.

While ScanVue can display characters outside of the usual ASCII range of 32–127, many libraries will not correctly handle those characters in normal text. Nominal mode transfers should be used to write text with the extended character set (128-255).

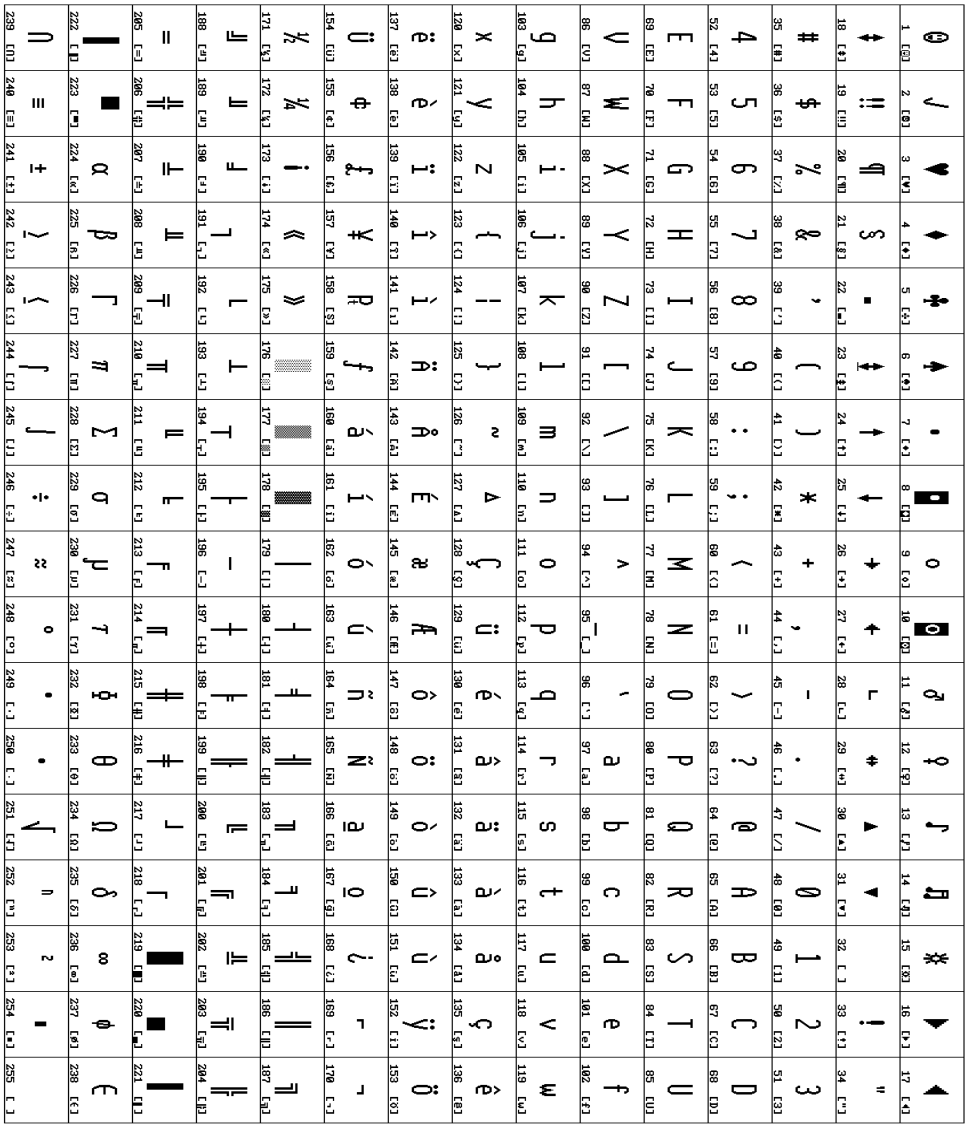


Table 8—Default ASCII Character Set For LCD Display

Vacuum Fluorescent (VF) Display Models

VF display models have the fixed built–in character set shown in Table 4. The empty code positions from A0-DF can be filled with one of the language character sets shown in Tables 5-8 [see section on ScanVue420 p50] with European as the default.

Table 9—Default ASCII Character Set for VF Display

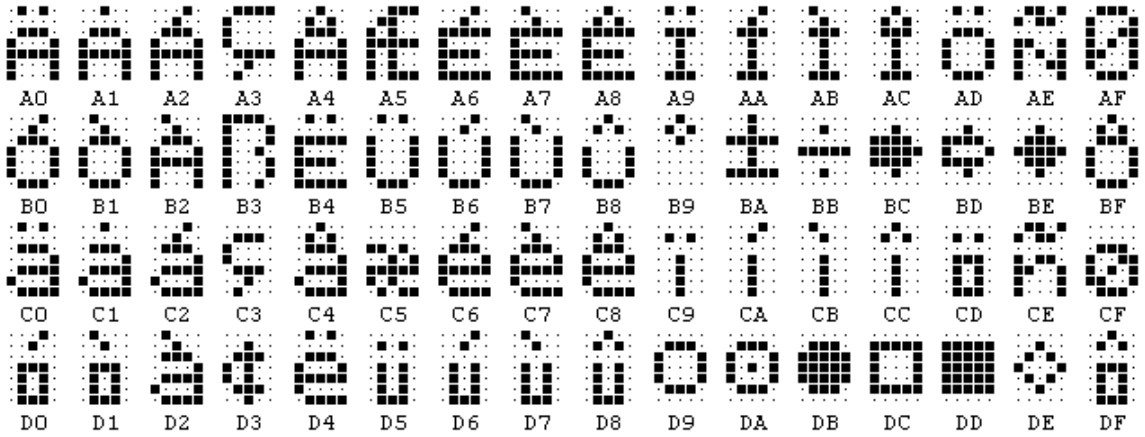

A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF

Table 10—European Character Set for VF Display

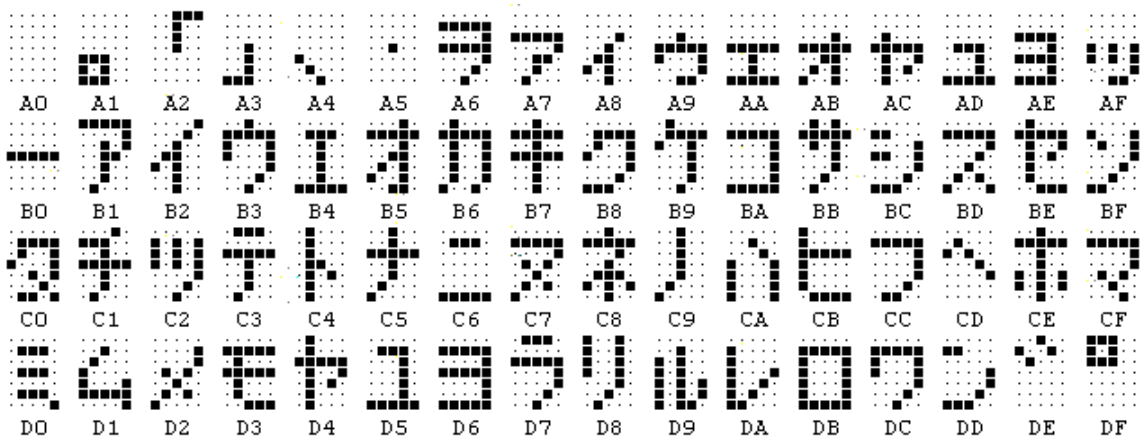

A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF

Table 11—Katakana Character Set for VF Display

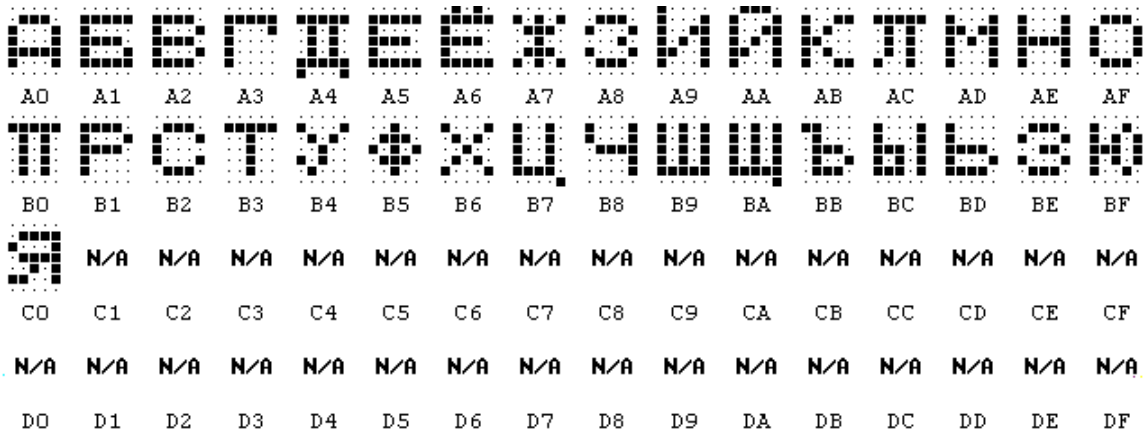

A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF

Table 12—Cyrillic Character Set for VF Display

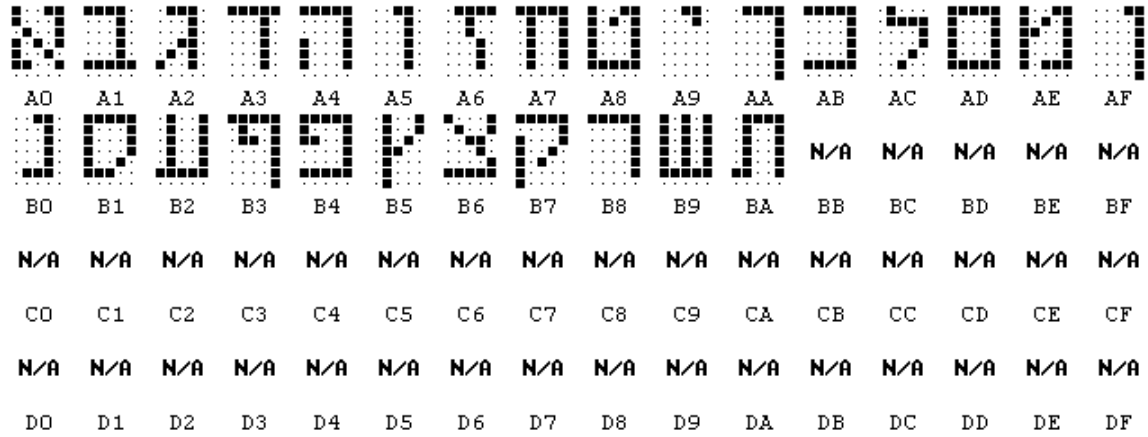

A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA N/A N/A N/A N/A N/A
BB BC BD BE BF
N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF

Table 13—Hebrew Character Set for VF Display

Appendix E—ProductInfo Protocol

Abstract: This specification describes a bi–directional message passing protocol called 'Product Information Protocol', that was created especially for retail store price–verifier applications. The protocol is designed to be generic and is not tied to any specific retail hardware device. Any network topology or configuration capable of using or connecting to TCP/IP will be able to support ProductInfo based applications.

The IEE Inc. ScanVue® and ShopVue® retail products use a sub–set of the ProductInfo protocol to meet their functionality requirements. The specific implementations for these products are covered in their respective user manual application programming interface (API) section.

IEE Inc.

7740 Lemona Avenue

Van Nuys, CA 91409–9234

Introduction

The ProductInfo Protocol provides a network–based messaging system whereby a client can obtain item price and description information about specific products. This information can be in any form such as text, graphic images, sound or combinations. The protocol will be submitted as an RFC for the Internet community.

Protocol Types

There are two forms of the protocol: trivial and nominal. The trivial version consists purely of <NUL> terminated text sent from the client to the host, or from the host to the client. From the client, it is a product query; from the host it is a text response. This may not support all the features of any particular device, so nominal mode must be used for advanced features.

The trivial and nominal cases can be distinguished by examination of the first byte; in trivial mode it will always be a printable ASCII character—in nominal it will be zero (unless you are sending individual packets in excess of 16MB). When a trivial–mode message is received by the server it is interpreted as a product query; it optionally contains the client's identification and white space preceding the product code. When received by the client, it is interpreted as a single, text response to a query. In either case, sessions are closed by the server.

In the nominal case, messages consist of a length, followed by a token, possibly followed by more information as specified by the length and the token.

Symmetry

The format is the same in both directions but the implementations at either end may or may not understand all the same tokens. In normal operation, the client opens a connection for each request, and keeps it open until the server instructs the client to close it. The client can also wait for the server to open a socket, to allow asynchronous operation. Either side may act as client, or server, or both.

Errors

In the interest of robustness, both ends will accept any message whether defined or not— invalid messages are discarded. A maximum reasonable message length may be used as a means to detect implementation bugs that could result in loss of synchronization; such errors terminate the connection. If the client detects a loss of synchronization it may send an error token following re–establishment of the connection in order to log the error on the server. If the server detects this condition, it can log it directly.

Following a query, the client may choose to take an error action if it receives nothing from the server within a defined timeout period.

Status Requests

The server can make capability queries and/or mode changes before, after, or in lieu of sending any response. The client may send capability messages regardless of whether the key name is known to the server; the server retains this information. When the server needs to know the value of one of these capabilities, it consults this retained information. If it is not known, a capability query may be sent and the server may wait a moment for a reply to be received. This reply will asynchronously update the server's information, and the value should be found there by a subsequent lookup following the brief interval required for the client to respond to the query. If it remains undefined, it can be assumed that the client declined to respond, most probably because that capability name is not known to it.

Client Mode Changes

The notion of a mode setting is to allow the server to select between optional behaviors or parameters in the client. If the server wants the client to adopt a certain mode setting, it sends the command. The client will do so, if applicable. If not, optionally, an error report may be generated in response. Mode values are set by using token–value pairs. Mode values can be queried by sending the mode name alone. Error reports are used primarily as a debugging tool. The string starts with an error number, optionally followed by white space and explanatory text.

Packet Types

Packet Types

Function	Token
Terminate connection	TERM
Capability query	CAPQ
Capability response	CAPR
Mode Set/Query	MDSQ
Mode Response	MODR
Error report	ERR_
Product query	PROD
Data	DATA
Registration	RGST
Event	EVNT

Data Types

Function	Token
No Operation	NOP_
Display Text	TEXT
Special Text[1]	SPEC
UNC	UNC_
URL	URL_
File	FILE

[1] The meaning of any Special Text is implementation–defined.

Nominal Mode Packets

General Every nominal–mode packet starts with a header containing the packet length and the token and may contain no other data. Packets are described below.

Terminate Connection: Requires only the length and the token.

Byte #	Length				Token			
	0	1	2	3	4	5	6	7
	0	**0**	**0**	**8**	**T**	**E**	**R**	**M**

Capability Query: The name of the capability to be queried is a string following the header. Capability names are case insensitive. The header for a SendUnitID capability query would be

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**19**	**C**	**A**	**P**	**Q**

The query string “SENDUNITID” and a terminating <NUL> immediately follow the header.

Capability Response: If the capability being queried is known, a response is sent. The name of the capability and its value is a string in token/value format following the header. The header for a SendUnitID response might be:

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**24**	**C**	**A**	**P**	**R**

The response string “SENDUNITID=TRUE” and a terminating <NUL> immediately follow the header. There can be white space on either or both sides of the equals sign. The value starts with the first non–white space character. Values can be empty. If the capability query is a zero–length string, ScanVue® responds with a <CR><LF> separated list of all supported capabilities and their types. If the query is a single axterix (‘*’), SVLite® will respond with a <CR><LF> separated list of modes and their values.

Mode Set/Query:

The name of the mode and its value are in a token/value string following the header. If the mode name alone is in the string, it is a query. Mode names are case–insensitive. The header for a SendUnitID query would be:

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**19**	**M**	**D**	**S**	**Q**

The query string "SENDUNITID" and a terminating <NUL> immediately follow the header. To set SendUnitID, the string would be (e.g.) "SENDUNITID=ON", and byte 3 would be 22 instead of 19.

Mode Response:

If the mode being queried of set is known, a response is sent. The name of the mode and its value is a string in token/value format following the header. The header for a SendUnitID response might be:

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**22**	**M**	**O**	**D**	**R**

The response string "SENDUNITID=ON" and a terminating <NUL> immediately follow the header. There can be white space on either or both sides of the equals sign. If the mode query is a zero–length string, ScanVue will respond with a <CR><LF> separated list of all supported modes and their types in the form "MODENAME=TYPE,SIZE" where TYPE is BOOLEAN, NUMBER, or STRING, and SIZE is the number of bytes in the STRING and only appears for STRING. If the mode query is the single–character string "*", ScanVue will respond with a <CR><LF> separated list of all supported modes and their current values.

Error Report:

Error reports are primarily intended for debugging purposes. If enabled, error reports are sent if, for example, the host queries a capability that the client does not accept. Such error strings are in the format of a number possibly followed by white space and explanatory text. There is no assignment of numbers to specific errors, but the value zero should be reserved for non–error conditions. An "OK" error response could be:

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**13**	**E**	**R**	**R**	**_**

The response string "0 OK" and a terminating <NUL> immediately follow the header.

Product Query:

The product query string immediately follows the header. The header for a request about product ABC would be*:*

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**12**	**P**	**R**	**O**	**D**

The query string "ABC" and a terminating <NUL> immediately follow the header. Mode settings can cause the client to send its unit ID and a <TAB> character before the product data, and/or add terminating <CR> and/or <LF> characters at the end of the query string.

Data:

Data can be one of several types. The data–type token immediately follows the packet–type token, and the data follows after that. The header for a data packet to display the word "hi" would be:

Byte #	0	1	2	3	4	5	6	7	8	9	10	11
	0	**0**	**0**	**15**	**D**	**A**	**T**	**A**	**T**	**E**	**X**	**T**

The data string "hi" and a terminating <NUL> immediately follow the header.

Registration:

Registration information consists of three sequential <NUL> terminated strings following the header. The first string is the unit ID, the second is the product–type identification, and the third is the IP address of the registering client. The header for a registration packet for "Entry", product type "SVLITE® 200102161901", and IP address of "10.0.0.227" would be:

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**79**	**R**	**G**	**S**	**T**

Each registration string and its terminating <NUL> immediately follows the header or the previous registration string. When a client receives a registration packet, it should respond by sending its own registration.

Event: A client can support up to 32 application specific events and can report the beginning and end of each event with a 100th second timer (which need not be supported). Events are mapped into a 32–bit word, in standard network order. Clients may allow the host to enable and disable individual events, or to select reports only for starting or ending events. The packet header for events from UnitID "Unit1" is:

Byte #	0	1	2	3	4	5	6	7
	0	**0**	**0**	**26**	**E**	**V**	**N**	**T**

Immediately following the header is: the 32–bit map of events that have started since the last report, the 32–bit map of events that have ended since the last report, a 32–bit timer value and a NUL terminated string that is the sending unit's UnitID. The bytes following the header if event zero started and event one ended at time 5 from UNIT1 would be:
0 0 0 1 0 0 0 2 0 0 0 5 85 78 73 84 49 0.

Client Requirements

Capabilities

The following capabilities must be supported by a client:

Timeout	Changing the client's protocol timeout value is supported.
SendUnitID	Sending the unit's ID can be enabled and disabled.
TrivialComm	Controls the default communication method used for queries.

Modes

The following modes must be supported:

Timeout	Hundredths of a second represented in a 32–bit value. This specifies the amount of time that a client will wait, after sending an initial query, before it terminates the connection to the server and displays an error message (if no other response has been received). The default is 300 (3 seconds).
TrivialComm	Controls the communication method used for queries. The default is TRUE. The response to a trivial–mode query need not be in trivial–mode.
SendUnitID	Controls the inclusion of the Unit ID in a trivial–mode query. The default is TRUE.
SendError	Controls the sending of error reports. The default is FALSE.
SendResponse	Controls the sending of responses to mode set commands. The default is FALSE.

Appendix F—Mounting Bracket Outline

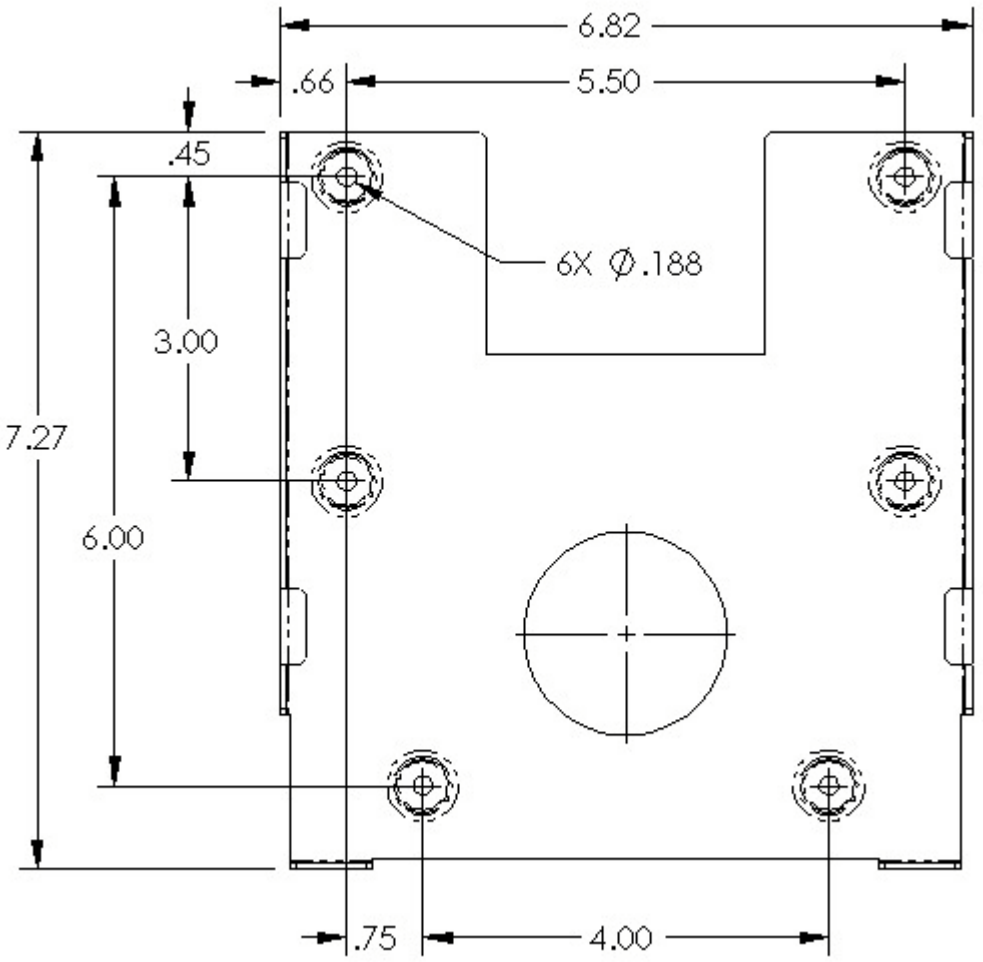


Fig 12—Wall Mount Bracket

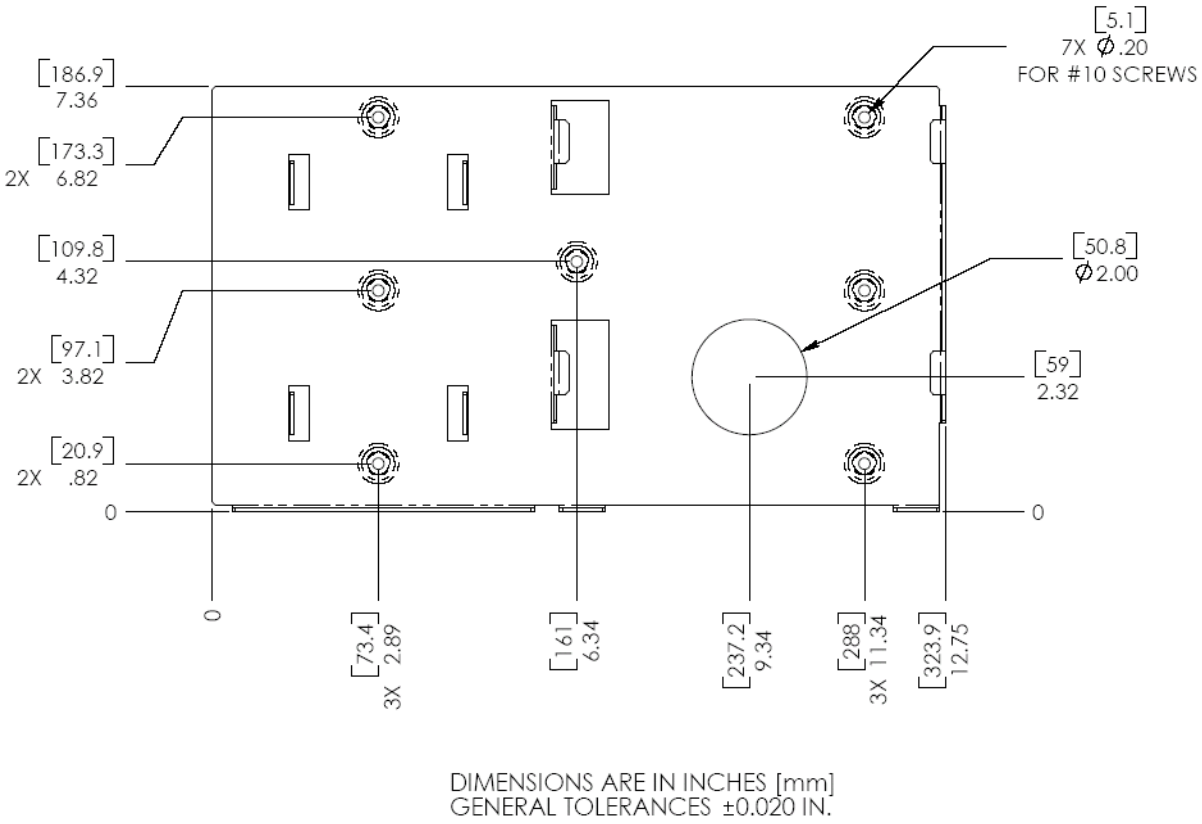


Fig 13—Co-Located Printer Outline Bracket

Appendix G—Application Notes

Creating Customer Messages on VF Display Models

The SVLite VF display model has 4 built-in language character sets, ASCII+European (default), ASCII+Cyrillic, ASCII+Hebrew and ASCII+Katakana that can be changed with a mode command from UnitConfig. The 4 extended language sets are shown in Appendix D.

There is no display formatting utility or language character conversion capability in this version, so messages (80 characters max including blank spaces) must be padded with spaces to center text or get it to appear as you want on the display. Non standard ASCII characters—this means almost all characters other than English upper and lower case and symbols, must be entered in hex code format.

Network Messages

Three network related customer messages generated within the price verifier can be changed by the user. The default messages are: **Checking….. one moment please** (Mode=msgChecking) and **Unavailable…..please try later** (Mode=msgUnavail) and **Welcome To IEE ScanVue Lite**……(mode=BannerText). Using the UnitConfig program, locate the mode (row) for the message you want to change, highlight the New Content box and enter the new message.

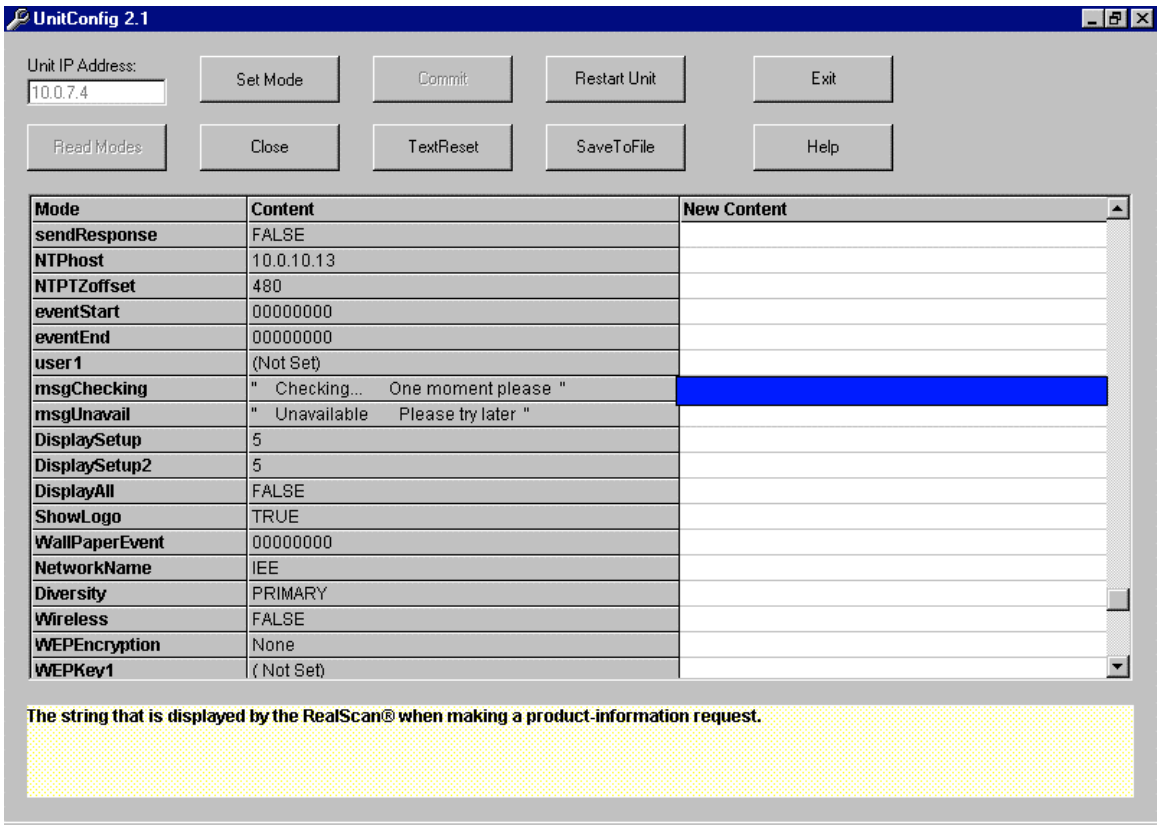


Fig 14—UnitConfig Screen, msgChecking Mode

When the message has been entered, click the buttons **Set Mode**, **Commit** and **Restart** in sequence. **Wait….** The unit will reboot but it takes about 1 minute before the initialize screen appears.

Example 1:
Changing **" Checking… one moment please "** into Spanish.

Highlight the New Content box in the msgChecking row as shown in Figure 30 above.

Enter **" Verificando… Un momento por favor"**

The quotes delineate the starting and ending positions of the message including spaces. In this message, there are 3 blanks at the start of the top line and 3 blanks at the end of the top line to center the word **Verificando...** The U in Un starts at the 1st position in the second line (as the display word wraps after the 20th character). As the second line is exactly 20 characters, there are no blanks inserted and the quotes close off the message after the word **favor**. This message entered as shown will display;

```
   Verificando...
Un momento por favor
```

Example 2:
Change **" Checking… one moment please "** into French.

Highlight the New Content box in the msgChecking row as shown in Figure 30 above.

Enter **"V\xC6rifier.. Un moment s'il vous pla\xCCt "**

In this example, 2 of the characters **é** (hex code \xC6) and **î** (hex code \xCC) are non standard and must be entered using the hex code method shown on page 43. When counting characters, hex codes occupy 1 character space position on the display. The message as entered will display as:

```
Vérifier.. Un moment
   s'il vous plaît
```

Greeting Message

Example:
Change the default English greeting message shown below to Spanish.

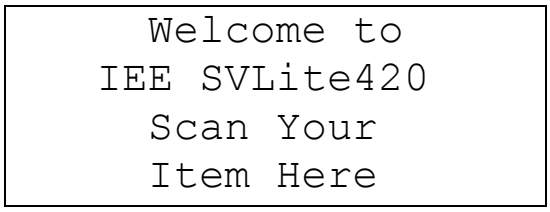


Fig 15—Default English Greeting Message

The Spanish text entry will be:

"**D\xC6*la*bienvenida******a*ScanVue*420*******Escudri\xCEe*su*******Art\xCAcule*aqu\xCA****"

where * denotes spaces. These are only used in the example for clarity and are not required when a message file is created. The hex codes count as 1 character space.

The new file will display as shown below.

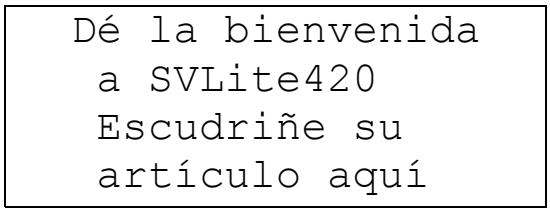


Fig 16—Spanish Greeting Message

Appendix H—Expanded Text API

With the standard text API, only one font size with a single foreground and background color can be displayed at a time. To support different fonts and colors on the screen at the same time, usually done within defined window areas (known as text boxes), an expanded text API is provided so that multiple text boxes can be defined, positioned and displayed with each text box having its own character property controls (font style, size and colors).

Up to eight (8) text boxes can be placed on the screen similtaneously including the default text box defined in the INI file. Each different font style used must be in a new text box, but each character within a single text box can have a different foreground and background color selected from the 256 colors available on the color palette. Any text box may be defined as "transparent" in which the text in that box will appear to float on top of the screen image or wallpaper with no visible text box background color. The whole text box is transparent not just individual characters.

API Notation

Items in braces { and } are optional. If not explicitly specified, the last value used for that item (the default value) is used. Items in angle brackets < and > are parameter values and are described in the text following.

Defining a Text Box

The following section describes how to define a text box. Up to 8 different text boxes can be defined. Once a text box is defined, it retains its properties until the unit is powered down or it is changed by another text message to that box even if its no longer being displayed.

Except for the default text box in the .INI file, text boxes are created at run-time by an escape sequence. A text box can be changed at any time, which deletes the old box and creates a new one. A text box can only contain a single font. Text boxes are identified by an ID # from 0-7.

Text boxes can be created with either trivial or nominal mode packets. Currently, no error reporting is provided.

<ESC>[D<ID>;<Fc>;{<FONT>};{<Fg>}{,<Bg>};{<Cr>}{,<Cc>};{<Ph>}{,<Pv>};{<TO>};

where:
<ID> Text Box ID #, ASCII 0-7. 0 is the .ini default text box

<Fc> Control Function—2 decimal digits, 0-9
00 = mixed text (text over an image)
01 = transparent text
02 = exclusive text

<FONT> The font file (specify in same way as in the .INI file)

<Fg> Foreground text color, ASCII 0 or 255
<Bg> Background text color, ASCII 255 or 0

<Cr> Count of text rows
<Cc> Count of text columns

<Ph> Horizontal pixel position of top left corner of box
<Pv> Vertical pixel position of top left corner of box
where top left corner of the display is 0,0

<TO> Text timeout

Important note: while the parameters in <> are optional, the semi-colons are not.
For example:

<ESC>[D1;2;fonts/bit16x32.gft;;1,2;16,16;;

defines text box 1 with the current default text colors and timeout. Those values come from the default text box (0).

Writing to a Text Box

Specify the text box ID # and the text to be displayed

<ESC>[<ID>;{<TEXT>}

The text string can include text color controls. Text sent without a box ID # will be inserted into the default text box, ID=0.

Text Timeout

When sent as part of a text message, causes the current text box to timeout (be removed from the screen). The TO value is the ASCII number of seconds

<ESC>[T<TO>;

Appendix J—Local Lookup Database

A feature of software versions after V1.7 allows loading a lookup file into the price-verifier that contains all the item barcodes and their associated description and price. Using a PLU file allows Scanvue to operate with a local database that will hold up to several thousand items. The PLU file may be updated as often as required—typically once a day to once a week.

Once this file is loaded (initially at boot up time and defaulted to check for updated versions every 5 minutes), the unit will run without having to access the network server database. If an item is not found in the PLU file, Scanvue will send the barcode request to the server. If the server is disconnected and no answer is provided, the request will timeout and return to the users main scan screen. The user is responsible for making any updates to the lookup file and placing it in a location where the Scanvue unit can find it. Scanvue must be connected to the network to find the scanvue.ini file and update the PLUFile.

If Scanvue has the 4 button option and a printer is connected, pressing the leftmost button will cause the displayed data to be printed to a label provided modes EventStart is set to 1 and Port2Function set to output.

Rules for Creating a PLUFile

1. The lookup file (PLUFile) to be loaded is specified in the [display] section of the scanvue.ini file by the keyword PLUFile.
   Ex: [display]
   PLUFile = lookup.txt the file lookup.txt will be loaded at boot time.

2. The PLU file size should be smaller than 512kbytes.
   The file size can be as big as 2Mbytes, however because a linear search algorithm is used, response time can be 5-6 secs with the larger files.

3. The PLU file is a text file with lines (items) separated by <NL> or <CR>. The PLU file is created by the user.

4. Each line consists of one or two barcode entries followed by the text description and price. A barcode entry is the ASCII barcode value followed by a tab character.
   Ex: 12345678 23456789 Wine Bottle Opener $7.95
   34567890 45678901 Bottle Cradle $17.99

5. Two new modes have been added to UnitConfig: PLUFile and SlideShowFile. These are “view only” modes that display the active or loaded PLU and Slideshow files in scanvue.ini. These modes cannot be changed from UnitConfig.

6. We do not recommend using the PLU file as a dynamic cache—it is not designed for this purpose.

Change history

Date	Page	Change
10/15/04		Initial Release
3/24/05	37	Added barcode to set scanner defaults to code128
2/23/06		Release V1.07
2/23/06	2	Added introduction to local database feature
2/23/06	28	Add PLUFile and SlideShowFile to supported modes
2/23/06	44	Add PLUFile to .ini definition
2/23/06	46	Add PLUFile and SlideShowFile to .ini example
2/23/06	65	Added Appendix J—local lookup file details
2/23/06	13	Replaced multiple Quickstart Guides with one.
2/23/06	61	Added co-located bracket outline drawing